MPLS-Based Best-Effort Traffic Engineering

A Thesis
Presented to
The Academic Faculty

by

Jerapong Rojanarowan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2005

MPLS-Based Best-Effort Traffic Engineering

Approved by:

Professor Henry L. Owen, Advisor
Electrical and Computer Engineering
Georgia Institute of Technology

Professor John A. Copeland
Electrical and Computer Engineering
Georgia Institute of Technology

Professor Randal T. Abler
Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: September 12, 2005

To my parents,

for their constant love.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitute to my advisor, Dr. Henry Owen, for his consistent dedication and guidance throughtout my study. I greatly appreciate his trust that allows me to freely explore, and his support for everything I need.

I am truely thankful to Dr. John Copeland, Dr. Randal Abler, Dr. Chuanyi Ji, and Dr. Jun Xu for their valuable time serving on my thesis defense committee, and providing useful comments and recommendations for this research.

I would like to thank Lawrence Sharp at the School of Industrial and Systems Engineering at Georgia Tech for providing the AMPL/CPLEX software license.

I am very grateful to Assumption University for giving me the opportunity to study at Georgia Tech. It is an invaluable experience.

I would like to express my appreciation to all members of the Antigravity and Teleportation Lab for their support and friendship. A special thanks goes to Dr. Bernd Koehler for all his help with ns-2 simulator.

I am greatly indepted for my parents, Suphon and Somchit. They give me absolute love and care. For that I cannot thank them enough.

Finally, I would like to give a special thanks to Dr. Napassavong for her unwavering love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

AQM Active Queue Management.

AR Access Router.

ARIS Aggregate Route-based IP Switching.

ATM Asynchronous Transfer Mode.

ATMARP ATM Address Resolution Protocol.

BR Backbone Router.

CoS Class of Service.

CR-LDP Constraint-based Routed Label Distribution Protocol.

CSR Cell Switch Router.

DiffServ Differentiated Services.

DMM-2 Disjoint Maxmin - 2 paths.

DMM-3 Disjoint Maxmin - 3 paths.

DMM-4 Disjoint Maxmin - 4 paths.

DOP-2 Disjoint Optimal - 2 paths.

DOP-3 Disjoint Optimal - 3 paths.

DOP-4 Disjoint Optimal - 4 paths.

DSCP Differentiated Services Codepoint.

ECN Explicit Congestion Notification.

ER-LSP Explicitly Routed Label Switched Path.

FEC Forwarding Equivalence Class.

GA Guaranteed Allocation.

GT-ITM Georgia Tech Internetworking Topology Model.

IETF Internet Engineering Task Force.

IntServ Integrated Services.

IP Internet Protocol.

ISP Internet Service Provider.

LANE LAN Emulation.

LSP Label Switched Path.

LSR Label Switched Router.

MA Mean Allocation.

MIRA Minimum Interference Routing Algorithm.

MPLS Multiprotocol Label Switching.

MPOA Multiprotocol Over ATM.

NHRP Next Hop Resolution Protocol.

OSPF Open Shortest Path First.

PA Proportional Allocation.

PHB Per-Hop Behavior.

POP Point of Presence.

QoS Quality of Service.

RED Random Early Detection.

REM Random Early Marking.

RIP Routing Internet Protocol.

RSVP Resource Reservation Protocol.

SLA Service Level Agreement.

SMM-2 Shortest Maxmin - 2 paths.

SMM-3 Shortest Maxmin - 3 paths.

SMM-4 Shortest Maxmin - 4 paths.

SNMP Simple Network Management Protocol.

SOP-2 Shortest Optimal - 2 paths.

SOP-3 Shortest Optimal - 3 paths.

SOP-4 Shortest Optimal - 4 paths.

SPF Shortest Path First.

TCP Transmission Control Protocol.

UDP User Datagram Protocol.

VCI Virtual Channel Identifier.

VPI Virtual Path Identifier.

WAN Wide Area Network.

WRR Weighted Round Robin.

SUMMARY

The objective of this research is to develop a multipath traffic engineering framework for best-effort traffic in Multiprotocol Label Switching (MPLS) networks so as to deliver more equal shares of bandwidth to best-effort users as compared to the traditional shortest-path algorithm. The proposed framework is *static* and the input to the traffic engineering algorithm is restricted to network topology. Performance evaluation of this framework is conducted by simulation using ns-2 network simulator.

In a multi-service capable network, some portion of the bandwidth is reserved for guaranteed services and the leftover portion is dedicated to best-effort service. This research examines the problem of traffic engineering for the remaining network bandwidth that is utilized by best-effort traffic where demands are not known *a priori*. This framework will result in making the limited available best-effort traffic bandwidth more equitably shared by the best-effort flows over a wide range of demands. Traditional traffic engineering research has not examined best-effort traffic.

CHAPTER I

INTRODUCTION

1.1 Motivation

Current Internet routing techniques are based on hop-by-hop routing that directs traffic along the shortest path. This technique is simple and scalable, but it has drawbacks. First, it causes unbalanced traffic distribution. Links on the shortest path are used and become congested, while other links not on the shortest path are underutilized. This can cause *"hot spots"* in the network. Next, it uses only one single best path from a source to its destination. Consequently, the total bandwidth a user received is potentially limited. Last, since user-perceived performance depends on how network bandwidth is shared by the routing algorithm, the shortest-path scheme, regardless of bandwidth sharing among contending flows on the links, delivers unfair performance among users on the network. Therefore, there is a possibility for traffic engineering to improve the performance of IP networks.

To provide such capability, the basic IP forwarding paradigm of present-day IP networks must be enhanced to support traffic engineering. The advent of multi-protocol label switching (MPLS) made this feasible by introducing the connection-oriented features of forwarding packets over arbitrary non-shortest paths.

There has been much research in the area of traffic engineering using MPLS. However, this previous research has focused on the guaranteed quality of service (QoS) traffic classes, which require admission control with resource reservation. Input traffic is well defined by its required minimum bandwidth and delay requirements. Thus, the amounts and types of traffic are available as inputs to traditional traffic engineering algorithms.

This research is targeted toward best-effort traffic engineering. There is no admission control and no guarantee of transmission for this best-effort traffic. All incoming calls are accepted to the network and there is no call blocking. This best-effort type of service works well in the existing Internet because most existing data applications are *elastic* in nature. These existing applications can tolerate some variations in network condition. A majority of traffic generated by these elastic applications is regulated by the transmission control protocol (TCP). This closed-loop congestion control mechanism regulates the sending rate by means of feedback control. The best-effort transmission rate is dependent on the network state, number of users sharing the bandwidth, and the traffic demands. All of these factors are dynamic in nature. Therefore, the transmission rate of best-effort traffic is not well defined. This makes the traditional traffic engineering path optimization techniques inapplicable since required inputs such as traffic demands are unknown for the best-effort case.

1.2 Dissertation Outline

The remainder of this thesis is organized as follows. Chapter 2 provides background information of the best-effort traffic, the MPLS architecture, the traffic engineering framework, and some of the recent related approaches to the traffic engineering in IP networks. Chapter 3 describes the proposed best-effort traffic engineering framework. Chapter 4 outlines the performance evaluation of the proposed framework. Chapter 5 presents the conclusions, research contributions, as well as directions for future research.

CHAPTER II

BACKGROUND

2.1 Overview

The Internet began as a research program in the early 1960s at the Advanced Research Projects Agency (ARPA), a unit within the Department of Defense (DoD). It was originally intended to provide connectivity to support a broad range of communications such as remote terminal access and file transfer over heterogeneous networks. Subsequently, the Internet gained interest and its use now ranges from small research communities to commercial sectors around the globe.

As the Internet becomes more popular, traffic on the network grows rapidly and approximately doubles each year [10]. In addition, there is a huge interest placing diverse types of emerging multimedia applications such as video conferencing, video-on-demand, and telemedicine on the Internet.

Despite the success of the Internet, its architecture and service model have remained the same. The Internet is still a datagram network offering a single class of service called *"best-effort"* service. This best-effort service works well in the existing Internet because most existing data applications such as e-mail, telnet, and file transfer are elastic in nature. These existing applications can tolerate some variations in network condition. However, multimedia applications are different than data applications. These real-time applications are typically less elastic - less tolerant to packet loss and delay variation - than traditional data applications. The tremendous traffic volume and novel requirements, as well as characteristics of multimedia applications pose a great challenge to the Internet.

Accordingly, over the course of the last decade, many novel technologies have been

proposed for the Internet to support a variety of communication services. Integrated Services (IntServ) [8] and Differentiated Services (DiffServ) [7] architectures have been proposed by the Internet Engineering Task Force (IETF) to offer different levels of services in addition to the dominating best-effort service. IntServ is a framework based on per-flow service with resource reservation. With this approach, the source must reserve resources before it can transmit to guarantee that its Quality of Service (QoS) requirements can be provided by the network. The Resource Reservation Protocol (RSVP) [9] is a signaling protocol that allows a sender to set up network resource reservation. It offers two service classes in addition to best-effort service: guaranteed service [42] and controlled-load service [50]. Since IntServ must maintain per-flow state information in routers, scalability is the primary problem in large-scale deployment. DiffServ, on the other hand, relies on per-class service. A Differentiated Services Codepoint (DSCP) field in the Internet Protocol (IP) header is used to select a Per-Hop Behavior (PHB) that defines how traffic belonging to a particular behavior aggregate is treated at a router. Besides best-effort service, there are two more service classes: expedited forwarding [11] and assured forwarding [17]. These new architectures accommodate the needs of new applications for guaranteed services and allow service providers to use service differentiation as a means to increase their revenue.

In addition, growing network traffic can cause network congestion - a condition when the offered load exceeds network capacity. Essentially, network congestion may result either from a shortage of bandwidth or inefficient traffic management that causes uneven traffic distribution. To overcome the former problem, providers are forced to overprovision their network bandwidth to keep pace with traffic demands. The latter problem results from the routing techniques in the IP network. The conventional routing techniques in the IP network are based on the shortest-path routing that maps traffic onto the shortest path, whereas the capacity of the links not on the

shortest-path is underutilized. This inefficient use of network bandwidth increases the operation cost of service providers that in turn inhibits their ability to compete in the market. Traffic engineering is a means to subdue this problem.

To provide such capability, the basic IP routing paradigm of the current Internet must be enhanced to support traffic engineering. The advent of Multi-protocol Label Switching (MPLS) made this feasible by introducing the connection-oriented features of forwarding packets over arbitrary non-shortest paths.

2.2 Internet Architecture

This section outlines the main characteristics of IP networks related to this research, including best-effort service model, shortest-path routing and destination-based forwarding techniques, and elastic traffic.

2.2.1 Best-Effort Service

Despite new enhancements and types of applications, the best-effort traffic with its elastic nature will dominate the Internet for years to come. In best-effort networks, there is neither admission control nor resource reservation for each connection because best-effort networks do not maintain information about each connection in routers along the path. All packets are serviced the same without delivery guarantee in the best-effort model. Packets may get dropped in the network because of lack of buffer space in the router or bit errors caused by noise during transmission. In addition, because packets are routed independently, they may take different paths and arrive at the destination out of order.

With no guarantee of packet delivery, additional functions in the host are required to provide reliable transmission for certain applications such as file transfer. Transmission Control Protocol (TCP) [35] on the end systems provides reliable packet delivery by retransmission after a failed delivery is detected by the lack of acknowledgement or a timeout. Other applications that do not require more than a best-effort

service can directly use this type of network. The User Datagram Protocol (UDP) [34] serves the need of these type of applications.

Since there is no admission control to limit the excess traffic, all connection requests are accepted to the network and there is no connection rejection. Clearly, when the offered load exceeds available network capacity, network congestion will occur. Therefore, the problem of congestion is inherent in the best-effort network because it lacks a traffic control mechanism.

Multimedia applications require intact data transmission in a timely manner. These requirements are not well suited to best-effort networks because retransmission is a feedback process that unavoidably incurs a time delay before reaction. This mechanism proves successful for elastic applications but not well for multimedia applications because of stringent time delay requirements. If the retransmitted packet arrives at the destination after its playback point, this packet will be discarded. From a network resources perspective, this situation is not desirable because network bandwidth is used for a wasted journey of the late retransmitted packet.

2.2.2 Internet Routing

Basically, the routing function in the Internet comprises two separate functional components: control and forwarding. A *control* function uses routing protocols such as Routing Internet Protocol (RIP) [30] and Open Shortest Path First (OSPF) [32] to exchange routing information with other routers. This routing information is necessary to build a forwarding table to be used by *forwarding* function. Particularly, these conventional routing protocols use a single metric, such as administrative weight or hop count, to characterize a link. To construct a forwarding table, the shortest-path algorithm is subsequently used to find the best path from a source to a destination based on this metric.

This technique is adequate for traditional applications that concern connectivity.

However, multimedia applications require a wide variety of QoS requirements. A link must be characterized by multiple metrics such as bandwidth, delay, or loss. As a result, QoS routing is a complex task to find a path that satisfies many constraints imposed by application requirements.

It is the forwarding function that transfers a packet from input to output across the router. When a router receives a packet, the forwarding function uses the information inside the header of this packet and the forwarding table to make a decision. Then, a packet is forwarded accordingly. In particular, a packet is forwarded based on its destination address, independent of its source address. This technique is advantageous to scalability because it aggregates numerous routes from different sources to the same destination into one single entry in a forwarding table.

While destination-based forwarding and shortest-path routing techniques are simple and scalable, shortest-path routing has several drawbacks. First, it causes unbalanced traffic distribution. Links on the shortest path are used and become congested, while other links not on the shortest path are underutilized. This can cause *"hot spots"* in the network. Next, it uses only one single best path from a source to its destination. Consequently, the toal bandwidth a user receives is potentially limited. Next, since user-perceived performance depends on how network bandwidth is shared by the routing algorithm, the shortest-path scheme, regardless of bandwidth sharing among contending flows on the links, delivers unfair performance among users on the network. Last, different sources with the same destination traversing to the same point will have the same path from that point. This limits the types of services the network can offer and makes traffic engineering impossible.

2.2.3 Elasticity

About 90% of the traffic in the Internet consists of elastic data applications mediated by TCP [37]. On top of the unreliable IP service, TCP provides end-to-end reliable

data delivery to applications. To provide reliability, TCP uses positive acknowledgement and timeout mechanisms. A byte-level *sliding window* operation in TCP allows a sender to transmit many packets before receiving acknowledgements. This technique greatly increases throughput because it prevents the network from being idle. In addition, since the number of outstanding bytes is limited by the window size advertised by a receiver, this technique also serves as an end-to-end *flow control* so that a sender does not transmit more bytes than a receiver can process. A receiver will send positive acknowledgements to identify the highest per-byte sequence number received correctly. If a packet is not acknowledged within a timeout period, this packet is retransmitted.

The primary reason that TCP is capable of delivering performance to numerous applications over a vast dynamic network environments is that TCP makes those mechanisms *adaptive*. The first concern is how long to wait for the acknowledgement in the timeout period. Since TCP operates end-to-end across many networks, a packet may traverse several hops through many routers and links. Links on the path may have different bandwidths, and routers may differ on processing capability. Moreover, the variable traffic level in each router dramatically varies the total time since a packet is transmitted until an acknowledgement arrives. If the timeout period is too short, the sender may unnecessarily retransmit a packet while the acknowledgement is on its way. On the other hand, timeout period that is too long will decrease throughput. TCP circumvents this problem by adaptively estimating the timeout period [35].

Flow control contributes to adaptivity as well. A receiver dynamically advertises a window size that indicates how many additional bytes it is ready to accept. According to an increased window size, the sender will transmit more packets. On the other hand, the sender will decelerate its transmission in response to a decreased window size. This mechanism makes the transmission rate of TCP adaptive and difficult to predict.

Another mechanism that makes TCP adaptive is *congestion control*. Unlike flow control, which is primarily intended to prevent the sender from overwhelming the receiver's buffer space, congestion control is designed to prevent overflowing the buffers of routers. TCP was originally designed without a congestion control scheme [35]. When congestion occurs in a network, packets experience long delay or get dropped. Without congestion control, a sender with a timeout and retransmission mechanism respond to congestion by introducing more copies of the same packet into the network. This clearly intensifies congestion and leads to congestion collapse. Congestion control mechanisms are necessary to decelerate the transmission of the sources when congestion occurs. In the late 1980s, congestion control mechanisms were introduced [18]. A new *congestion window* was introduced at the source. The maximum rate the source can transmit is limited to a minimum of the receiver's advertised window and its congestion window that is dynamically changed according to network congestion levels.

In summary, TCP, originally designed to work in adaptive environments, is a dynamic protocol. The transmission rate of the source regulated by closed-loop flow control and congestion control depends on the network condition, number of users, and traffic demands. All of these factors are dynamic in nature. As a result, the transmission rate of the source is not well defined.

2.3 QoS Routing

Multimedia applications have very stringent QoS requirements like bandwidth, delay, jitter, and reliability. To provide guaranteed performance for these applications, network resource reservation must be set up beforehand. However, resource reservation is not available in the current Internet. In addition, the shortest-path routing algorithm optimized for a single metric is not QoS-aware. Hence, packets may traverse a path that has insufficient resources. Packets taking different paths will experience

unpredictable delay and arrive at the destination out of order. Obviously, this type of application prefers connection-oriented service.

The emergence of RSVP provides a mechanism for reserving network resources. However, before the resource reservation can take place, QoS routing is needed to find a feasible path that has adequate resources for QoS requirements. Because these requirements impose a set of constraints on a routing problem, in some cases, they make QoS routing intractable.

A link must be characterized by multiple metrics. These metrics can be classified into three types: additive, multiplicative, and concave [48]. For an *additive* metric, e.g., delay, jitter, and cost, a path metric is the summation of the link metric on all links in this path. For a *multiplicative* metric, e.g., loss, a path metric is the multiplication of the link metric on all links in this path. For a *concave* metric, e.g., bandwidth, a path metric is the minimum of the link metric on all links in this path.

In routing problems, there are two types of constraints: link constraints and path constraints [26]. A *link constraint* is a link selection criterion that restricts links to be included in a path. This type of constraint usually involves the concave metric. For example, to support a minimum required bandwidth, all links on a path must have available bandwidth greater than the bandwidth requirement. Links with insufficient available bandwidth are simply excluded from path computation. On the other hand, the *path constraint* is more complicated. It is an end-to-end restriction on a path. This type of constraint usually involves additive and multiplicative metrics. For instance, to support the end-to-end delay requirement, a path must have less delay than the requirement. In fact, a routing problem subject to two or more independent path constraints is NP-complete [48].

2.4 MPLS

MPLS largely evolved from the need to integrate IP with Asynchronous Transfer Mode (ATM). ATM switches were attractive because they could provide high-speed forwarding capability over wide area networks. The *overlay network*, where high-speed ATM switches may be used to interconnect IP routers, was typically used in service provider's backbone networks. However, IP and ATM are two dramatically different architectures with different addressing structures and control schemes. IP is connectionless, whereas ATM is based on connection-oriented transmission.

To integrate IP with ATM, some mapping functions are required. The IETF has published mapping protocols such as ATM Address Resolution Protocol (ATMARP) [25] and Next Hop Resolution Protocol (NHRP) [28] to address the mapping problems. The ATM forum has proposed Multiprotocol Over ATM (MPOA) [2] integrating LAN Emulation (LANE) [1] with NHRP to transport layer-3 protocols across ATM networks. All of these server-based overlay approaches are complex and have a scalability problem.

To overcome the scalability problem, many *label-switching* approaches, including IP Switching [33], Tag Switching [36], Aggregate Route-based IP Switching (ARIS) [49], and Cell Switch Router (CSR) [20], are utilized by many vendors. The IETF has standardized MPLS [41] to avoid functional incompatibilities brought by different label-switching techniques. Besides integrating IP with ATM, MPLS provides many useful capabilities to IP networks such as forwarding speed, QoS, and traffic engineering.

Recall that routing comprises control and forwarding functions. In traditional routing techniques used in the Internet, these two functions are closely blended. To add a new capability such as multicast routing requires changes in both functions. MPLS separates these two functions. New routing techniques can be added to the control function without affecting the forwarding function. MPLS displaces the

destination-based forwarding paradigm with a label-swapping forwarding paradigm. Unlike conventional destination-based forwarding schemes where the table lookup is used to find the longest-prefix match, the label switching scheme performs table lookup for an exact label that can be achieved in a single memory access. This reduction in memory access time results in faster forwarding speed.

A label is a short, fixed-length, locally significant identifier. A label encoding is link layer-specific. For a native label-switching technique like ATM and Frame Relay, a label can be readily encoded in the Virtual Path Identifier (VPI)/Virtual Channel Identifier (VCI) fields, and the Datalink Connection Identifier (DLCI) field, respectively. Other link-layer techniques that cannot accommodate a label in their header use a four-byte *shim header* [40]. The header includes a 20-bit label that contains the actual value of the label. A three-bit experimental use is reserved but can be used for Class of Service (CoS) as in Tag Switching. One bottom-of-stack bit is used to identify the last entry in the label stack. Finally, a time-to-live field carries an eight-bit counter.

Every label corresponds with a Forwarding Equivalence Class (FEC). An FEC is considered as a group of packets that has the same path forwarding requirements. In the Internet where packets are forwarded based on destination address independently of source address, a destination address is an example of an FEC. Because MPLS supports multiple types of FECs, it provides a wide range of forwarding granularities ranging from per-destination to per-application [27].

An ingress Label Switched Router (LSR) of an MPLS domain divides incoming packets into many FECs. Then, it assigns a label to a packet corresponding to its FEC. Inside an MPLS domain, a labeled packet is forwarded based on the label-switching table that maps an incoming label to its outgoing interface and outgoing label. The incoming label is replaced by the outgoing label before a labeled packet is forwared to the next LSR. An egress LSR is responsible for removing a label from a

packet before it leaves an MPLS domain.

A Label Switched Path (LSP) is a concatenation of labels. Since a mapping between labels is maintained by each LSR, the first label completely determines the LSP. MPLS directs the flows of packets along the predetermined LSPs across the network based on the label switching. In MPLS, there are two alternative LSP selection mechanisms: hop-by-hop and explicit routing. A *hop-by-hop* path is calculated based on the normal layer-3 routing information. With an *explicit routing* mechanism, the path is completely assigned by the originator, independent of layer-3 routing. There are two methods to set up an explicit route: Constraint-based Routed Label Distribution Protocol (CR-LDP) [19] and extensions to Resource Reservation Protocol (RSVP) [4].

2.5 Traffic Engineering

Traffic engineering is the process of performance optimization of an operational network or fulfilling some policy objectives [5]. Performance objectives are both *resource-oriented* and *traffic-oriented*. The resource-oriented objectives are related to the network operation. This type of objective includes optimization of resource utilization such as minimizing network congestion by efficient bandwidth management. On the other hand, traffic-oriented performance objectives are associated with customers including QoS enhancement of traffic flows such as loss, delay, and throughput. Essentially, traffic engineering is the task of mapping traffic flows onto a physical network to meet certain objectives.

Supporting traffic engineering clearly requires a connection-oriented feature that is able to set up arbitrary non-shortest paths across the network. However, the limitation of traffic engineering in IP networks is due to the destination-based forwarding scheme. When two traffic flows to the same destination merge, it is impossible to split and route them over different paths.

The dawn of MPLS allows traffic engineering in IP networks. The Explicitly Routed LSP (ER-LSP) with optimization objectives is the primary tool for traffic engineering in MPLS. With ER-LSP, ingress LSRs are able to control traffic distribution in a network to meet performance objectives. However, the exact algorithm for determining the ER-LSP is not specified in the IETF.

2.6 Related Work

There has been much research in the area of traffic engineering and quality of service in the Internet. In this section, an overview of this previous research is presented. Note that all path-selection techniques reviewed in this section assume that traffic demands are known *a priori*. However, this is not true for best-effort traffic where traffic demands are not well-defined. Hence, all of these path-selection techniques are not applicable for best-effort traffic.

2.6.1 QoS in Best-Effort Networks

Wydrowski and Zukerman [52] present a price-based load control scheme to deliver differentiated QoS in best-effort networks. To achieve this goal, two mechanisms are needed. A *source flow control* and an *active queue management* (AQM). The first element regulates how much traffic it should send based on the congestion level. Thus, it represents a bandwidth buyer that can control the amount of bandwidth purchased. The AQM will estimate the congestion level and send a congestion signal back to the source by dropping packets or explicit congestion notification (ECN). With a congestion signal representing a price per bit of transmission, AQM is a bandwidth seller that manages the price. By using a pricing mechanism, users are charged according to the load they place on the network. All users transmitting on a path contribute to the total demand; the same congestion signal is sent back to all users competing for this path since user-perceived performance with transmission rate x is different depending on its utility function $U(x)$. In response, users trying to maximize

their *net benefit* $\mathrm{B} = U(x) - xc$, where c is the cost, will transmit differently depending on their utility functions. Thus, differentiated QoS can be achieved. However, there are no simulation results of this scheme given in their paper.

With *queue-based* AQM algorithms such as tail-drop and Random Early Detection (RED) [12], the number of congestion signals depends on the queue size. To increase the number of congestion signals, the queue size must also increase. This leads to an unnecessarily long delay time. The authors suggest using *rate-based* AQM algorithms, including Random Early Marking (REM) [3] and GREEN [51], that estimate congestion based on flow arrival rate. Therefore, the congestion signal is sent before the queue builds up.

2.6.2 QoS Routing for Bandwidth Guarantees

To guarantee an end-to-end performance of multimedia applications, a path with bandwidth guarantee is required. The *widest-shortest path* algorithm [16] chooses the minimum hop count path among all feasible paths having residual bandwidth larger than the required bandwidth. The residual bandwidth is used to break the tie. The modified Dijkstra's algorithm with two distance functions, hop count and bandwidth, can be used. The hop count is used first to select the next node. If there are many nodes with the same hop count, the one with the largest bandwidth is selected. The algorithm terminates when the destination is attained.

Another technique, called *shortest-widest path* [48], selects the largest bandwidth path among all feasible paths having residual bandwidth larger than the required bandwidth. The hop count is used to break the tie. To find such a path, Dijkstra's algorithm is applied twice. First, links with residual bandwidth less than the required bandwidth are pruned from the network. Next, the largest bandwidth path is calculated by using Dijkstra's algorithm on the reduced network. Then, links having bandwidth less than this largest bandwidth are pruned. Last, a minimum hop path

is selected from this reduced network by using Dijkstra's algorithm.

Since the widest-shortest path algorithm gives preference to limiting resource usage, it performs better at heavy loads, whereas the shortest-widest path algorithm is better at light loads because it prefers a non-congested path that is useful to balance network load [29].

2.6.3 Traffic Engineering in Datagram IP Networks

Another technique presented by Fortz and Thorup [13] is the traffic engineering scheme in IP networks based on optimizing link weight settings for intradomain routing protocols such as OSPF. The goal is to avoid overloaded links. The idea is that, in shortest-path routing, the shortest path from a source to a destination is calculated based on link weight. Thus, by changing link weight settings, the routes as well as the traffic loads on the links can be changed. This enables traffic engineering in the IP network. However, finding an optimal weight setting in OSPF is NP-hard. The authors propose a local search heuristic technique to find the optimal solution.

There are three steps. First, traffic demand and network topology are obtained from measurement or the Simple Network Management Protocol (SNMP). Next, with this information, the external traffic engineering server optimizes the setting of static link weights to meet performance objectives. Last, routers are reconfigured with these new weight settings.

Let the network be represented by the directed capacitated graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ the set of directed links between them. Every link $(i, j) \in E$ has the associated capacity C_{ij}. Let $K \subseteq V \times V$ represent the set of all source-destination (S-D) pairs in the graph. Define D^k as the traffic demand of S-D pair $k = (s^k, d^k) \in K$. Total traffic load l_{ij} on link (i, j) is defined as

$$l_{ij} = \sum_{k \in K} f_{ij}^k \tag{1}$$

where f_{ij}^k is the traffic flow of S-D pair k on link (i, j). The cost of link (i, j), defined

as the convex piecewise linear function of the total traffic load on that link, is

$$\phi_{ij}(l_{ij}) = \begin{cases} l_{ij} & \text{if} \quad 0 \leq l_{ij} < \frac{C_{ij}}{3} \\ 3l_{ij} - \frac{2}{3} & \text{if} \quad \frac{C_{ij}}{3} \leq l_{ij} < \frac{2C_{ij}}{3} \\ 10l_{ij} - \frac{16}{3} & \text{if} \quad \frac{2C_{ij}}{3} \leq l_{ij} < \frac{9C_{ij}}{10} \\ 70l_{ij} - \frac{178}{3} & \text{if} \quad \frac{9C_{ij}}{10} \leq l_{ij} < C_{ij} \\ 500l_{ij} - \frac{1468}{3} & \text{if} \quad C_{ij} \leq l_{ij} < \frac{11C_{ij}}{10} \\ 5000l_{ij} - \frac{19468}{3} & \text{if} \quad \frac{11C_{ij}}{10} \leq l_{ij} < \infty \end{cases} \tag{2}$$

The optimal routing can be formulated as follows.

$$\text{minimize} \quad \Phi = \sum_{(i,j)\in E} \phi_{ij}(l_{ij}) \tag{3}$$

such that

$$\sum_{j:(i,j)\in E} f_{ij}^k - \sum_{j:(j,i)\in E} f_{ji}^k = \begin{cases} +D^k & \text{if} \quad i = s^k \\ -D^k & \text{if} \quad i = d^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i \in V, \forall\, k \in K \tag{4}$$

$$l_{ij} = \sum_{k\in K} f_{ij}^k \qquad \forall\, (i,j) \in E \tag{5}$$

$$\phi_{ij} \geq l_{ij} \qquad \forall\, (i,j) \in E \tag{6}$$

$$\phi_{ij} \geq 3l_{ij} - \frac{2C_{ij}}{3} \qquad \forall\, (i,j) \in E \tag{7}$$

$$\phi_{ij} \geq 10l_{ij} - \frac{16C_{ij}}{3} \qquad \forall\, (i,j) \in E \tag{8}$$

$$\phi_{ij} \geq 70l_{ij} - \frac{178C_{ij}}{3} \qquad \forall\, (i,j) \in E \tag{9}$$

$$\phi_{ij} \geq 500l_{ij} - \frac{1468C_{ij}}{3} \qquad \forall\, (i,j) \in E \tag{10}$$

$$\phi_{ij} \geq 5000l_{ij} - \frac{19468C_{ij}}{3} \qquad \forall\, (i,j) \in E \tag{11}$$

$$f_{ij}^k \geq 0 \qquad \forall\, (i,j) \in E, \forall\, k \in K \tag{12}$$

The optimal objective value Φ_{OPT} is used to compare with the objective value

Φ_{OSPF} obtained from a proposed local search heuristic. It is found that the maximum gap between Φ_{OPT} and Φ_{OSPF} is less than a factor of 5000. In fact, Φ_{OSPF} is 5000 times greater than Φ_{OPT} when traffic demand approaches infinity. The reason is that the optimal routing is considered as the ideal scheme that establishes one or more explicit paths between each source-destination pair and distributes arbitrary amounts of traffic on each path. In contrast, OSPF is able to balance load equally over multiple equal cost paths. The authors have found that good weight settings can perform within a few percentage points of optimal routing in a real AT&T backbone network. In comparison with standard weight settings including inverse capacity and hop count, this technique can support a 50-110% increase in traffic demand. In addition, good weight settings are not very sensitive to the exact details of the objective function. Furthermore, only a few changes are needed in the case of failures or topology changes. The existing weights continue to perform well even after a link failure [13].

Vutukury and Garcia-Luna-Aceves [46] proposed a traffic engineering system based on minimum delay routing. The minimum delay routing is actually the optimum routing [6] with the objective function to minimize the total delay, which is the sum of the delay on each link. It is a non-linear programming problem. The calculation is done offline. It is assumed that the traffic demands are known.

Let the network be represented by the directed capacitated graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ the set of directed links between them. A full-duplex link is considered as two separate directed links. Every link $l = (i, j) \in E$ has the associated capacity C_{ij}. Let $K \subseteq V \times V$ represent the set of all source-destination (S-D) pairs in the graph. Define D^k as the traffic demand of S-D pair $k = (s^k, d^k) \in K$ and P^k as the set of paths for S-D pair $k \in K$. Let $P = \cup P^k$. Define x^p as the traffic flow on path $p \in P$. The total traffic flow of all S-D pairs on

link (i, j) is f_{ij}. The minimum delay routing is formulated as follows.

$$\text{minimize} \sum_{(i,j)\in E} \frac{f_{ij}}{C_{ij} - f_{ij}} \tag{13}$$

such that

$$f_{ij} = \sum_{p\in P:(i,j)\in p} x^p \quad \forall\,(i,j) \in E \tag{14}$$

$$f_{ij} \leq C_{ij} \quad \forall\,(i,j) \in E \tag{15}$$

$$\sum_{p\in P^k} x^p = D^k \quad \forall\, k \in K \tag{16}$$

$$x^p \geq 0 \quad \forall\, p \in P \tag{17}$$

After the optimal solution of this problem $\hat{x}^p$, which is the set of traffic flows, is obtained, the LSP or virtual circuit is set up for each traffic flow. Each source distributes traffic over multiple paths in correspondence with the flow rates using a mechanism such as Weighted Round Robin (WRR). However, the authors argue that MPLS is complex and they propose new forwarding techniques using *routing parameters* [6] to overcome the drawback of MPLS. These new forwarding techniques can be implemented within the current IP network to provide connection-oriented features without needing support from MPLS. The routing parameter ϕ_{ij}^k that must be maintained in a routing table is the fraction of all traffic arriving at node i, destined for node d, and forwarded on link (i, j). It is defined as

$$\phi_{ij}^d = \frac{f_{ij}^d}{\sum_{j:(i,j)\in E} f_{ij}^d} \in [0, 1] \tag{18}$$

where f_{ij}^d is the traffic flow on link (i, j) destined for node d. Note that

$$\sum_{j:(i,j)\in E} \phi_{ij}^d = 1 \tag{19}$$

Every router is configured with these parameters. When a router receives a packet, it selects an outgoing link based on the routing parameters. However, since WRR

forwards the packet regardless of packet size, it cannot allocate bandwidth fairly. Thus, it is difficult to achieve the fractions defined by routing parameters. The authors proposed new forwarding mechanisms that are variants of Deficit Round Robin (DRR) [43].

2.6.4 Traffic Engineering in MPLS Networks

Wang et al. [47] modeled the traffic engineering problem as an optimization problem, with the objective of minimizing maximum link utilization. It is assumed that the traffic demands are known. They present two models, one linear programming for *bifurcated* demand and one integer programming for *non-bifurcated* demand case with only one explicit route allowed for each demand. While the former case can be solved in polynomial time, the latter case is NP-hard. The authors proposed four heuristic algorithms to solve this problem.

Let the network be represented by the directed capacitated graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ the set of directed links between them. Every link $l = (i, j) \in E$ has the associated capacity C_{ij}. Let $K \subseteq V \times V$ represent the set of all source-destination (S-D) pairs in the graph. Define D^k as the traffic demand of S-D pair $k = (s^k, d^k) \in K$ and P^k as the set of paths for S-D pair $k \in K$. The fraction of S-D pair k's demand provided by link (i, j) is defined as x_{ij}^k. Let α_{ij} be the link utilization of link (i, j). The maximum link utilization α is defined as

$$\alpha = \max_{(i,j)\in E} [\alpha_{ij}] \tag{20}$$

$$\alpha \geq \alpha_{ij} \qquad \forall\, (i, j) \in E \tag{21}$$

The bifurcated demand case can be formulated as follows.

$$\text{minimize} \quad \alpha \tag{22}$$

such that

$$\sum_{j:(i,j)\in E} x_{ij}^k - \sum_{j:(j,i)\in E} x_{ji}^k = \begin{cases} +1 & \text{if} \quad i = s^k \\ -1 & \text{if} \quad i = d^k \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V, \forall\, k \in K \tag{23}$$

$$\sum_{k\in K} D^k x_{ij}^k \leq \alpha C_{ij} \qquad \forall\, (i,j) \in E \tag{24}$$

$$x_{ij}^k \in [0,1] \qquad \forall\, (i,j) \in E, \forall\, k \in K \tag{25}$$

$$\alpha \geq 0 \tag{26}$$

The optimal solution $\hat{x}_{ij}^k > 0$ determines a path set P^k for every $k \in K$ as well as the fraction associated with each path $p \in P^k$. Then, an LSP is set up for each $p \in \cup P^k$. Note that x_{ij}^k is a real value between 0 and 1. Then, the demand is split over multiple paths. If the demand is split on the packet level, the traffic will arrive out of order at the destination. This will degrade throughput of TCP-based applications.

In the non-bifurcated demand case, x_{ij}^k is a binary value of either 0 or 1. This will force the demand to take exactly one path. The non-bifurcated demand case is modeled as follows.

$$\text{minimize} \quad \alpha \tag{27}$$

such that

$$\sum_{j:(i,j)\in E} x_{ij}^k - \sum_{j:(j,i)\in E} x_{ji}^k = \begin{cases} +1 & \text{if} \quad i = s^k \\ -1 & \text{if} \quad i = d^k \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V, \forall\, k \in K \tag{28}$$

$$\sum_{k\in K} D^k x_{ij}^k \leq \alpha C_{ij} \qquad \forall\, (i,j) \in E \tag{29}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall\, (i,j) \in E, \forall\, k \in k \tag{30}$$

$$\alpha \geq 0 \tag{31}$$

This problem is NP-hard and the authors proposed four approximation algorithms to obtain a near-optimal solution.

The first proposed algorithm is the *Shortest Path* (SP), where the cost metric a_{ij}^k is the inverse of the link capacity, i.e.,

$$a_{ij}^k = \frac{1}{C_{ij}} \quad \forall\, (i,j) \in E, \forall\, k \in K \tag{32}$$

The *Minimum Hops* (MH) is very similar to SP except that the metric is the hop count. That is

$$a_{ij}^k = 1 \quad \forall\, (i,j) \in E, \forall\, k \in K \tag{33}$$

The *Shortest-Widest Path* (SWP) chooses the path with the largest bottleneck bandwidth. The minimum hop is used to break the tie. The last one is the *LPF-Based Re-Routing* (LPF-RR), which uses the linear programming formulation as a starting point. Then, it takes out the split demands and reroutes them with a single explicit route. This process will be done sequentially until α cannot be improved.

The results show that the split demand is only 2-8% of the total demand. Thus,

the number of re-routed demands is very small. LPF-RR performs best, and its maximum link utilization is very close to that of LPF. It is also found that LPF-RR can reduce the maximum link utilization below 40%.

Girish et al. [14] formulated the traffic engineering problems, including constraint based routing, admission control, and rerouting problems, mathematically and showed that they are NP-complete. However, they do not propose any algorithms to solve such problems.

The *constraint-based routing* problem is modeled as a binary integer programming. Hence, there is only one path for each S-D pair. The basic idea is to map the traffic demands from all S-D pairs to the physical network while satisfying a set of constraints and optimizing an objective function. The proposed objective function is to minimize the total administrative cost of all links.

Let the network be represented by the directed capacitated graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ the set of directed links between them. A full-duplex link is considered as two separate directed links. Every link $l = (i, j) \in E$ has the associated capacity C_{ij} and administrative cost per unit flow a_{ij}. Denote p_{ij} as the maximum allocation multiplier or oversubscription factor of link (i, j). Let $K \subseteq V \times V$ represent the set of all source-destination (S-D) pairs in the graph. Define D^k as the traffic demand of S-D pair $k = (s^k, d^k) \in K$ and P^k as the path for S-D pair $k \in K$. Let $P = \cup P^k$ be the set of all paths. Denote h^k as the allowed maximum number of hops for S-D pair $k \in K$. The binary desicion variable x_{ij}^k, indicating whether demand from S-D pair k is routed over link (i, j) or not is defined as

$$x_{ij}^k = \begin{cases} 1 & \text{if demand from S-D pair } k \text{ is routed over link } (i, j) \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

The constraint-based routing problem can be formulated as

$$\text{minimize} \quad \sum_{(i,j)\in E} a_{ij} \sum_{k\in K} D^k x_{ij}^k \tag{35}$$

such that

$$\sum_{j:(i,j)\in E} x_{ij}^k - \sum_{j:(j,i)\in E} x_{ji}^k = \begin{cases} +1 & \text{if} \quad i = s^k \\ -1 & \text{if} \quad i = d^k \\ 0 & \text{otherwise} \end{cases} \quad \forall\, k \in K \tag{36}$$

$$\sum_{k\in K} D^k x_{ij}^k \leq p_{ij} C_{ij} \qquad \forall\, (i,j) \in E \tag{37}$$

$$\sum_{(i,j)\in E} x_{ij}^k \leq h^k \qquad \forall\, k \in K \tag{38}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall\, (i,j) \in E, \forall\, k \in K \tag{39}$$

The *admission control* problem, dealing with whether to accept or reject a new request from an S-D pair, say n, with demand D^n can be formulated similar to the previous case with some modifications. To decide whether the network has sufficient capacity for the new request, for each link (i,j), the capacity of the link C_{ij} is changed to the residual capacity of the link R_{ij}. Once the request is accepted, the path with minimum cost must be determined. Hence, only the cost incurred by this new request is considered in the objective function. The admission control problem is formulated as follows.

$$\text{minimize} \quad \sum_{(i,j)\in E} a_{ij} D^n x_{ij}^n \tag{40}$$

such that

$$\sum_{j:(i,j)\in E} x_{ij}^n - \sum_{j:(j,i)\in E} x_{ji}^n = \begin{cases} +1 & \text{if} \quad i = s^n \\ -1 & \text{if} \quad i = d^n \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V \tag{41}$$

$$D^n x_{ij}^n \leq p_{ij} R_{ij} \qquad \forall\, (i,j) \in E \tag{42}$$

$$\sum_{(i,j)\in E} x_{ij}^n \leq h^n \tag{43}$$

$$x_{ij}^n \in \{0,1\} \qquad \forall\, (i,j) \in E \tag{44}$$

If there is no feasible solution, the LSP is rejected. Otherwise, it is routed according to the optimal solution $\hat{x}_{ij}^n$. For every link (i,j), when $\hat{x}_{ij}^n = 1$ this link belongs to the path.

The *rerouting* problem is needed for many reasons, including link and node failures, preemption, and load balancing. When a link (u,v) fails, we set the constraints for this link so that there would be no traffic on that link, i.e., $x_{uv}^k = 0\ \forall k \in K$. Similarly, for a node g failure, all links connected to this node will be set to bear no traffic, i.e., $x_{ig}^k = x_{gi}^k = 0\ \forall\ (i,g),(g,i) \in E, \forall k \in K$. Then, all affected S-D pairs are rerouted similar to the admission control problem. When a link or node recovers from a failure, a re-optimization is needed. This is achieved by relaxing the associated constraints and computing the new optimal solution. However, to avoid interruption, minimizing the number of rerouting S-D pairs is desirable.

Another bandwidth guaranteed routing algorithm scheme, called *Minimum Interference Routing Algorithm* (MIRA), is proposed by Kodialam and Lakshman [23]. It

is an online algorithm that handles incoming LSPs one at a time. The goal is to find a bandwidth guaranteed path for each incoming LSP, requiring a path bandwidth of D from a source a to a destination b, that optimally utilizes network resource. The information about source-destination (S-D) pairs, considered as the potential of future demands, is known. However, the demand itself is unknown until it arrives at the ingress LSR, where its QoS attribute, bandwidth requirement in particular, for the LSP is derived. Other metrics such as delay and loss can be converted to an effective bandwidth [15].

MIRA is a complicated algorithm, but its basic idea is that a newly routed LSP must follow a route that does not interfere too much with a route that may be critical for a future demand. In other words, the demand that may load certain critical links and hence block future demand is postponed. The maxflow between a source-destination pair is defined as an upper bound on the total bandwidth that can be routed between this pair. When an LSP from a pair is routed, the maxflow of the other pairs may be reduced. The amount of maxflow reduction is defined as *interference*. The algorithm will find the minimum interference path for an LSP that maximizes the weighted sum of the maxflow of every other pair.

Let the network be represented by the directed capacitated graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ the set of directed links between them. A full-duplex link is considered as two separate directed links. A matrix R is an m vector of residual capacities. Every link $l = (i, j) \in E$ has the associated residual capacity R_{ij}. Let $|V| = n$ and $|E| = m$. A matrix M is an $n \times m$ dimensional node-link incident matrix. Each row corresponds to a node and each column corresponds to a link. Hence, a link (i, j) will have a $+1$ in row i and a -1 in row j. Let $K \subseteq V \times V$ represent the set of all source-destination (S-D) pairs in the graph. Define a scalar θ^{sd} as the maxflow of S-D pair (s, d). Let e^{sd} be an n vector with a -1 in position s and a $+1$ in position d. Define a scalar α^{sd} as the administrative weight associated with

S-D pair (s,d). The m dimensional traffic flow vector of S-D pair $k=(s,d)$ is f^{sd} and the corresponding traffic flow of this S-D pair on link (i,j) is f_{ij}^{sd}. The problem to route the bandwidth demand of D from S-D pair (a,b) in MIRA is formulated as follows.

$$\text{maximize} \sum_{(s,d)\in K-(a,b)} \alpha^{sd}\theta^{sd} \tag{45}$$

such that

$$Mf^{sd} = \theta^{sd}e^{sd} \qquad \forall\,(s,d)\in K-(a,b) \tag{46}$$

$$Mf^{ab} = -De^{ab} \tag{47}$$

$$f^{sd}+f^{ab} \leq R \qquad \forall\,(s,d)\in K-(a,b) \tag{48}$$

$$f^{sd} \geq 0 \qquad \forall\,(s,d)\in K \tag{49}$$

$$f^{ab} \in \{0,D\}^m \tag{50}$$

The optimum solution of this problem is $\hat{f}^{ab}$. However, this problem is NP-hard. The authors propose a heuristic algorithm based on the shortest path with appropriate link weight settings. Once the link weights are determined, the shortest path algorithm is used to find a path for this LSP.

The proposed solution first assumes $D=0$ and independently calculates the maxflow $\hat{\theta}^{sd}$ for every $(s,d)\in K-(a,b)$. The objective value is now equal to

$$\sum_{(s,d)\in K-(a,b)} \alpha^{sd}\hat{\theta}^{sd} \tag{51}$$

However, this objective value can only be reduced when a non-zero demand is routed over links that belong to the mincut of any S-D pair. Let a set of critical links C_{sd} of S-D pair (s,d) be a set that includes every link on its mincuts. Accordingly, a weight w_{ij} of a link (i,j) is defined as

$$w_{ij} = \sum_{(s,d):(i,j)\in C_{sd}} \alpha^{sd} \tag{52}$$

This link weight approximately measures how much the objective value is reduced if a link is used to route the new demand. It can also be viewed as a link's degree of interference with other S-D pairs. It is clear that a path with minimal reduction of the objective value (minimum interference) is preferred. Before finding the weighted shortest path, any link that has residual bandwidth less than demand D is pruned from the graph. Links with zero weight, which do not belong to any mincuts, are assigned with a small positive number to ensure that the minimum hop path will be chosen.

Basically, it is a bandwidth guaranteed routing technique based on the shortest-path routing. Therefore, the bandwidth requirement for each flow is known. In contrast, the best-effort traffic has no information about the amount of demand. In addition, by relying on the shortest-path routing, only one best path is used to route traffic.

Based on the idea of MIRA, another online technique, called *Profile-Based Routing*, is proposed by Suri et al. [45]. Like MIRA, it is assumed that S-D pairs are known in advance. However, this algorithm further assumes that traffic profiles of those S-D pairs are also known. The traffic profile is used as an estimator for future demand that can be obtained by measurement or derivation from Service Level Agreement (SLA). Each traffic profile is defined by a quadruple $(class, s, d, B)$, where $class$ identifies the traffic class, s is the source, d is the destination, and B is the expected bandwidth demand. Similarly, an incoming request is defined by a quadruple (id, s, d, b), where id identifies the request ID for this particular S-D pair, s is the source, d is the destination, and b is the required bandwidth demand. It is assumed that the ingress s is able to classify the traffic class of every incoming request.

The algorithm consists of two steps: preprocessing and online. In the *preprocessing* step, the traffic profile is used to solve a multi-commodity flow problem, where each

traffic class is considered a distinct commodity $k \in K$, to determine the bandwidth allocation on the links for each traffic class. Solving the multicommodity flow problem for all traffic profiles may lead to an infeasible solution if the network capacity is insufficient. To avoid infeasibility, an *artificial link*, having infinite capacity and infinite cost, is added to every S-D pair. The network is represented by a directed capacitated graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ the set of directed links between them. Every link $l = (i, j)$ has the associated capacity C_{ij}. Let $\acute{E}$ be the set of artificial links. Define the link cost p_{ij} as

$$p_{ij} = \begin{cases} 1 & \text{if} \quad (i,j) \in E \\ \infty & \text{if} \quad (i,j) \in \acute{E} \end{cases} \tag{53}$$

Define a net demand for commodity k at node i B_i^k as

$$B_i^k = \begin{cases} +B^k & \text{if} \quad i = s^k \\ -B^k & \text{if} \quad i = d^k \\ 0 & \text{otherwise} \end{cases} \tag{54}$$

Let f_{ij}^k be an amount of traffic flow of commodity k on link (i, j). The multicommodity flow problem is formulated as follows.

$$\text{minimize} \sum_{(i,j) \in E \cup \acute{E}} (p_{ij} \sum_{k \in K} f_{ij}^k) \tag{55}$$

such that

$$\sum_{j:(i,j) \in E \cup \acute{E}} f_{ij}^k - \sum_{j:(j,i) \in E \cup \acute{E}} f_{ji}^k = B_i^k \quad \forall\, i \in V, \forall\, k \in K \tag{56}$$

$$\sum_{k \in K} f_{ij}^k \leq C_{ij} \quad \forall\, (i,j) \in E \cup \acute{E} \tag{57}$$

$$f_{ij}^k \geq 0 \quad \forall\, (i,j) \in E \cup \acute{E}, \forall\, k \in K \tag{58}$$

The optimum solution $\hat{f}_{ij}^k$ of this problem is used to allocate bandwidth to each traffic class k on link (i, j). With this information, a reduced graph for each traffic

class k is constructed. For each traffic class k, every link (i, j) is assigned a bandwidth of f_{ij}^k.

In the *online* algorithm, the residual bandwidth R_{ij}^k for each traffic class k on link (i,j) is recorded and updated when an LSP is routed. To route an LSP with bandwidth demand b^k from source s^k to destination d^k, an online algorithm will prune links that do not have enough residual bandwidth, i,e., links that have $R_{ij}^k < b^k$. Next, the minimum hop path from s^k to d^k is selected from this reduced graph. If one exists, the LSP will be accepted and the residual bandwidth of this path will be updated, otherwise it will be rejected.

The *Alternate Path Routing Algorithm* is proposed by Subramanian and Muthukumar [44]. It comprises two phases: prepocessing and online. In the *preprocessing* phase, an existing shortest-path routing algorithm is used for a period of time. Then, the traffic flow pattern is recorded on every link. This information is used to identify the *critical links* in which the amount of traffic flow is equal to the capacity of those links.

The *online* phase is divided into two planes: control and forwarding. In the *control* plane, all paths from each S-D pair are computed. First, the shortest-path routing is used to calculate the *shortest-path cost* (SPC) for all S-D pairs. Then, the *alternate paths*, which are paths having a cost less than or equal to $(1+\alpha)SPC$, are computed. The *alternate path parameter* $\alpha \in [0, 1]$ is an optimization variable chosen by the administrator. Next, the number of critical links present in each path is calculated. The paths are then sorted according to the number of critical links. The *primary path* is the path that has the lowest number of critical links in it. The path cost is used to break the tie. The rest of them are called *additional paths*. The *forwarding* plane that introduces the admission control works as follows. The incoming traffic is routed on the primary path. If the primary path becomes congested, the first additional

path is used. If the first additional path is also congested, the other non-congested additional path is used. If all paths are congested, the traffic is blocked.

This is a multipath routing scheme. However, the traffic is not split over multiple available paths but rather is alternately routed over the best non-congested path. The admission control algorithm does not take the bandwidth requirement of the incoming flow into consideration. In addition, traffic flow is only accepted when a non-congested path is available. In real networks where such a path rarely exists, all traffic flows are always rejected.

2.6.5 Fairness

In traditional traffic engineering algorithms and QoS routings, traffic demand volumes are specified and the network is required to find paths that satisfy the necessitated constraints for these demand volumes. In contrast, traffic demands in best-effort networks are elastic and not well-defined. Traffic demands can sustain the changes in bandwidth and utilize total bandwidth assigned to them. One of the essential problems in best-effort networks is how to assign bandwidth to the traffic flows in a fair fashion.

In the simplest case, the demands are assigned bandwidth equally. We assign each traffic flow $k \in K$ a bandwidth of x^k where

$$x^k = \alpha \qquad \forall\, k \in K \tag{59}$$

However, this simplest assignment might leave some links not fully utilized which is not the case for elastic traffic. The elasticity of these demands will expand their rate and fill the pipe. Then we have

$$x^k \geq \alpha \qquad \forall\, k \in K \tag{60}$$

Note that some traffic flows can receive more bandwidth *without* the expenses of other's shares. This leads to the notion of max-min fairness [6].

2.6.5.1 Max-min Fairness

An allocation of bandwidths is max-min fair if it is not possible to increase the assigned bandwidth of any flow at the expenses of the assigned bandwidth of flows that have larger shares. More formally, a vector of rates $x = (x^k, k \in K)$ is max-min fair if it is feasible for the following constraints

$$\sum_{k \in K} a_{ij}^k x^k \leq C_{ij} \qquad \forall\ (i,j) \in E \tag{61}$$

$$x^k \geq 0 \qquad \forall\ k \in K \tag{62}$$

where a_{ij}^k is a given routing matrix defined as

$$a_{ij}^k = \begin{cases} 1 & \text{if link } (i,j) \text{ belongs to the path of traffic flow } k \\ 0 & \text{otherwise} \end{cases} \tag{63}$$

and x^k for every $k \in K$ cannot be increased without decreasing x^{k^*} for some $k^* \in K$ for which $x^{k^*} \leq x^k$ within the feasible domain.

The algorithms to compute max-min fair share for a fixed set of paths was proposed in [6]. It is called "filling procedure". The procedure is as follows.

1. Assign all traffic flows with zero bandwidth at the beginning.
2. Increase the rates of all flows equally until some link capacity limits are reached.
3. Freeze the rates of those flows traveling the saturated links.
4. Continue until it is not possible to increase the rates.

The algorithm to compute a max-min fair share in a multipath case was developed by Koehler in his thesis [24].

2.6.5.2 Proportional Fairness

Another fairness criterion proposed by Kelly [21] is called "proportional fairness". It is defined as follows. An allocation of rates $x = (x^k, k \in K)$ is proportional fair if it

satisfies the following feasible constraints

$$\sum_{k \in K} a_{ij}^k x^k \leq C_{ij} \qquad \forall\, (i,j) \in E \tag{64}$$

$$x^k \geq 0 \qquad \forall\, k \in K \tag{65}$$

where a_{ij}^k is a given routing matrix defined as

$$a_{ij}^k = \begin{cases} 1 & \text{if link } (i,j) \text{ belongs to the path of traffic flow } k \\ 0 & \text{otherwise} \end{cases} \tag{66}$$

and for any feasible allocation y the average changes are zero or negative

$$\sum_{k \in K} \frac{y^k - x^k}{x^k} \leq 0 \tag{67}$$

It has been shown in [21] that if there exists an allocation of rate x that solve the following problem

$$\text{maximize} \qquad \sum_{k \in K} \log(x^k) \tag{68}$$

such that

$$\sum_{k \in K} a_{ij}^k x^k \;\leq\; C_{ij} \qquad \forall\, (i,j) \in E \tag{69}$$

$$x^k \;\geq\; 0 \qquad \forall\, k \in K \tag{70}$$

then x is proportional fair.

The main difference between max-min and proportional fairness is that the max-min fairness is subject only to the constraints imposed by link capacity limits. The proportional fairness introduces the economic reason by applying a logarithmic utility function for each flow that represents the law of diminishing returns. The objective is to maximize the overall utility of rate allocation $x = (x^k, k \in K)$.

2.6.5.3 α-proportional Fairness

An alternative approach is proposed by Mo and Walrand [31] to generalize the idea of proportional fairness. It is called "α-proportional fairness". The following class of

utility functions is proposed

$$U_k(\alpha, x^k) = \begin{cases} (1-\alpha)^{-1}(x^k)^{1-\alpha} & \text{if} \quad \alpha \neq 1 \\ \log(x^k) & \text{if} \quad \alpha = 1 \end{cases} \tag{71}$$

for $\alpha \geq 0$. Note that it reduces to proportional fairness when $\alpha = 1$, and max-min fairness when $\alpha = \infty$. The α-proportional fairness is defined as follows. An allocation of rates $x = (x^k, k \in K)$ is α-proportional fair if it satisfies the following feasible constraints

$$\sum_{k \in K} a_{ij}^k x^k \leq C_{ij} \qquad \forall\, (i,j) \in E \tag{72}$$

$$x^k \geq 0 \qquad \forall\, k \in K \tag{73}$$

where a_{ij}^k is a given routing matrix defined as

$$a_{ij}^k = \begin{cases} 1 & \text{if link } (i,j) \text{ belongs to the path of traffic flow } k \\ 0 & \text{otherwise} \end{cases} \tag{74}$$

and for any feasible allocation y the average changes are zero or negative

$$\sum_{k \in K} \frac{y^k - x^k}{(x^k)^\alpha} \leq 0 \tag{75}$$

It has been shown in [21] that if there exists an allocation of rate x that solve the following problem

$$\text{maximize} \qquad \sum_{k \in K} U_k(\alpha, x^k) \tag{76}$$

such that

$$\sum_{k \in K} a_{ij}^k x^k \;\leq\; C_{ij} \qquad \forall\, (i,j) \in E \tag{77}$$

$$x^k \;\geq\; 0 \qquad \forall\, k \in K \tag{78}$$

then x is α-proportional fair. This fairness definition is a useful way to compare different fairness policies.

All of the previous definitions of fairness have one common feature. They can only be computed once a fixed set of paths is given. Thus, the allocation process consists of two phases. First, a set of paths is determined. The set of paths can be shortest-path, k-shortest, or k-disjoint path sets. Last, for a given set of paths, the fairness definition is applied. The rates of traffic flows depend enormously on a given path set. In this thesis, we propose a combination of path finding and rate allocation techniques that yields an optimal balance between fairness and maximum network throughput.

CHAPTER III

PROPOSED FRAMEWORK

3.1 Overview

Koehler [24] proposed a best-effort traffic engineering framework in MPLS networks to improve network utilization. This earlier work was evaluated using a *flow-based* simulator. Since the flow-based simulator assumes infinite queue lengths, this simulator does not capture the effect of dropped packets and feedback mechanisms of TCP. To reflect more realistic behavior of best-effort traffic, and to create a simulator and baseline for evaluating the new framework, Koehler's old framework was re-examined using the *packet-based* ns-2 network simulator [39] [38]. The objectives of this new research include developing a new best-effort traffic engineering framework with an objective of delivering more equal shares of bandwidth to users. Performance evaluation of this new framework is conducted by simulation using the ns-2 network simulator.

3.2 Framework Architecture

One of the main purposes in designing the proposed new best-effort traffic engineering framework is to acquire a *simple* framework that is able to improve performance over the shortest-path routing. The framework is *static* and independent of network condition, and the input to the traffic engineering algorithm is limited to network topology. As a result, routing instability is avoided. In an adaptive scheme, where the congestion level of the links is taken into account, there is a feedback between routing algorithm and flow pattern. This introduces a possibility of routing oscillations that will deteriorate the performance. The simplicity of the framework also allows easy

deployment.

In this framework, a multipath routing scheme is employed because of the shortcomings of shortest-path routing, as stated earlier. The proposed framework is a *centralized* model, assuming that a traffic engineering server has information of network topology and link bandwidths. This server performs *offline* computation of optimized paths and rates for all available source-destination pairs. Subsequently, this information is distributed and installed at each label-switching router (LSR). A label-switching path (LSP) is populated in correspondence with each optimized path obtained from the traffic engineering algorithm. Traffic from a source-destination pair is directed over these predefined LSPs inside the MPLS domain. To enforce and guarantee that an LSP will receive its precomputed optimum rate along its way to the destination, a scheduling scheme is needed. This framework assumes that every LSR in the domain is equipped with a chosen scheduling discipline. For each path, all LSRs, except the penultimate one, are to be configured with the same scheduling parameter.

3.2.1 Components

To support the proposed new best-effort traffic engineering operation, the following tasks are required

- *Path selection*

  After network topology information $G(V, E, C)$ is obtained, for every source-destination pair $(s, d) = k \in K$, a path set P^k is calculated. This good path set P^k is used to distribute traffic load from source s to destination d. A corresponding label-switching path (LSP) is generated for each path in the set.

- *Rate calculation*

  An optimum rate x^{kp} is subsequently computed for each path $p \in P^k$ and $k \in K$.

Total flow rate x^k of a source-destination pair k is equal to $\sum_{p \in P^k} x^{kp}$. Clearly, the total flow rate on a certain link must not exceed the link bandwidth.

- *Traffic splitting*

  To support load balancing, a source node of a source-destination pair k distributes its traffic proportionally over multiple paths $p \in P^k$. All other nodes except the source are not required to perform this function. A proportional fraction ϕ^{kp} of traffic of k on path p is defined as $\frac{x^{kp}}{\sum_{p \in P^k} x^{kp}}$. Note that $\sum_{p \in P^k} \phi^{kp} = 1$.

- *Scheduling*

  A scheduling technique such as weighted fair queueing (WFQ) is required at all LSRs to guarantee that the flow receives its rate x^{kp} on every link along the path.

3.3 Fairness and Network Utilization

One basic problem in best-effort networks is the routing problem. Routing is a process of determining paths from a source to every destination in the network. Traditional routing algorithms are concerned with network connectivity. A weight associated with each link, called a "metric", such as delay and hop count, is used to characterize a link. A shortest path based on the chosen metric is calculated and used to forward data from a source to its destination. Internet routing protocol such as Routing Internet Protocol (RIP) and Open Shortest Path First (OSPF) use shortest path routing regardless of bandwidth sharing among contending flows. In a best-effort network, where a majority of applications are elastic in nature, user perceived-performance depends on how network bandwidth is shared by the routing algorithm. An efficient routing algorithm must be able to divide the available bandwidth of the network to all competing users as fairly as possible and to maximize the total allocated bandwidth to the users. However, these two objectives inherently contradict one another.

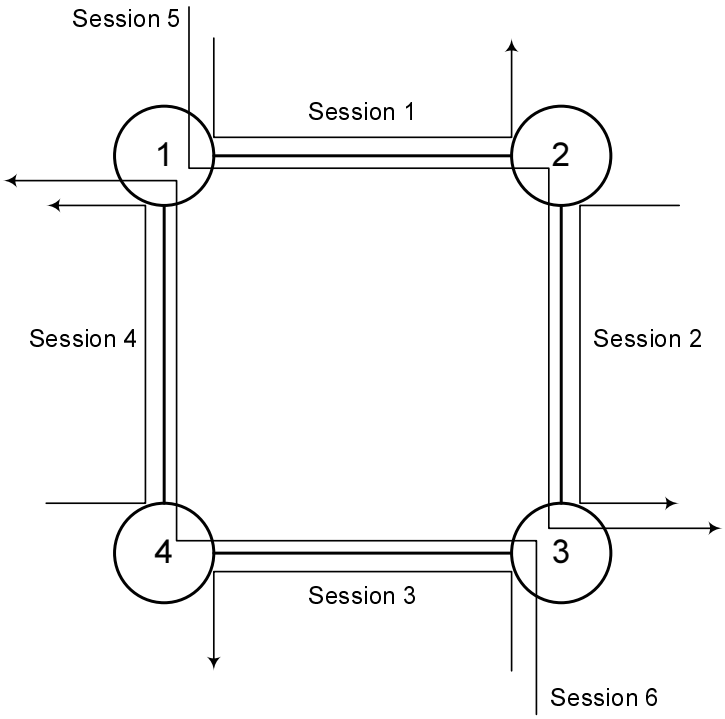

Figure 1: Maximum bandwidth allocation vs. fairness.

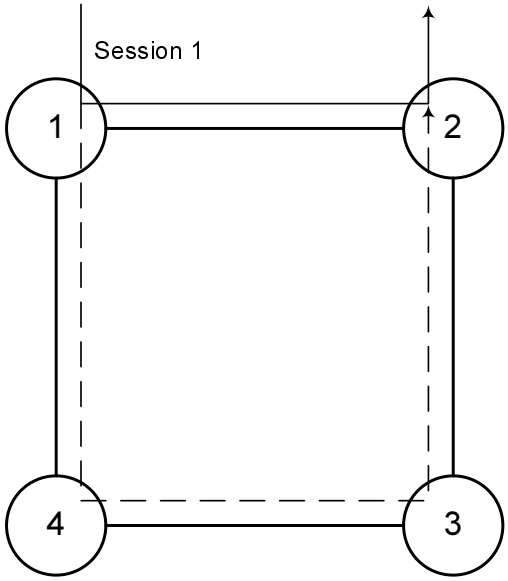

Figure 2: Maximum bandwidth allocation via multipath routing.

Consider the example illustrated in Figure 1. There are six sessions, each with traffic of 1 Mbps. All links have a capacity of 1 Mbps. A maximum throughput that this network can offer is 4 Mbps by allocating 1 Mbps to each one-hop session and blocking off the two-hop sessions. In this way, the total allocated bandwidth critirion is maximized, whereas the fairness criterion is not met because two two-hop sessions sacrifice their bandwidth portion for maximizing total bandwidth allocation. To achieve fairness, the total bandwidth maximization must be relaxed, allowing each session an equal portion of 0.5 Mbps. The total bandwidth in this case is 3 Mbps.

In Figure 2, assume that there is one session from node 1 to node 2; all links have a capacity of 1 Mbps. If routing from node 1 to node 2 is calculated based on the shortest-path algorithm, the maximum bandwidth is 1 Mbps via path 1→2. However, if node 1 is allowed to split its traffic along another longer path, namely, 1→4→3→2, the maximum bandwidth is doubled to 2 Mbps.

3.4 Bandwidth Allocation Algorithms

Maximum throughput and fairness objectives conflict with each other. In other words, if we allocate bandwidth such that the maximum throughput is achieved, some flows might have to sacrifice their bandwith portion and receive a zero bandwidth. On the other hand, if we assign bandwidth equally to all flows, we will not meet the maximum throughput objective which means the network capacity is not efficiently utilized. In order to avoid both scenarios, we have to find an optimal balance betweeen the maximum throughput and fairness objectives.

In this research, we proposed three methods to allocate bandwidth among competing users, namely guaranteed allocation, mean allocation, and proportional allocation.

In *guaranteed allocation*, all competing flows will be assigned at least a guaranteed bandwidth of $B > 0$. This technique consists of two phases. In the first phase, the objective is to find a maximum non-zero bandwidth B such that all flows will receive

bandwidth equally. None of these flows will receive a zero bandwidth. However, there will be some leftover bandwidth after the first phase. During the second phase, the leftover bandwidth is allocated to all flows such that the maximum throughput is obtained.

In *mean allocation*, with the results obtained from the guaranteed allocation, we will divide the flows into two separate sets: the set of flows that have bandwidth less than the average bandwidth, and the set of flows that have bandwidth more than the average bandwidth. There are two phases. First, we will increase the traffic flows in the lower bandwidth set as close to the average bandwidth as possible. Next, the leftover capacity is assigned to all flows such that the total throughput is maximized.

In *proportional allocation*, we will assign each flow the guaranteed bandwidth porportional to its maxflow. This is a three-step process. In the first step, we will determine the maxflow of each flow. The maxflow D^k is the maximum throughput of the flow when there are no other flows in the network. Thus, this maxflow represents the upper bound of throughput of this flow. In the next step, we will find the maximum guaranteed bandwidth for all flows proportional to their maxflow. During the last step, all leftover bandwidth is assigned to every flow such that the maximum throughput is obtained.

3.4.1 Path-Based Formulation

Let a network be represented by the directed capacitated graph $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ the set of directed links between them. A full-duplex link is considered as two separate directed links. Every link $(i, j) \in E$ has an associated capacity C_{ij}. Let $|V| = n$ and $|E| = m$. Let $K \subseteq V \times V$ represent the set of all source-destination (S-D) pairs in the graph and P^k be a set of paths for S-D pair $k = (s^k, d^k) \in K$. Assume that there is a set $N_A \subseteq N$ of active nodes and a set $N_P \subseteq N$ of passive nodes, with $N_A \cap N_P = \emptyset$ and $N_A \cup N_P = N$. An active node is

an ingress (or egress) node supplying (or consuming) traffic whereas a passive node is a transit node relaying traffic flows. The traffic flow of S-D pair k on path p is x^{kp}. In a single path case, the traffic flow $x^k = x^{kp}$ because there is only one path for each S-D pair. In a multipath case, the traffic flow x^k is the sum over all given paths. We have $x^k = \sum_{p \in P^k} x^{kp}$.

The maximum allocated bandwidth problem can be formulated as follows.

$$\text{maximize} \qquad \sum_{k \in K} \sum_{p \in P^k} x^{kp} \tag{79}$$

such that

$$\sum_{k \in K} \sum_{p \in P^k} a_{ij}^{kp} x^{kp} \;\leq\; C_{ij} \qquad \forall\, (i,j) \in E \tag{80}$$

$$x^{kp} \;\geq\; 0 \qquad \forall\, k \in K, \forall\, p \in P^k \tag{81}$$

where a_{ij}^{kp} is a given path routing matrix defined as

$$a_{ij}^{kp} = \begin{cases} 1 & \text{if link } (i,j) \text{ belongs to the path } p \in P^k \text{ of S-D pair } k \\ 0 & \text{otherwise} \end{cases} \tag{82}$$

The optimal solution of this problem is $\hat{x}^k \geq 0, \forall\, k \in K$. As shown in the earlier example, $\hat{x}^k = 0, \exists\, k \in K$, which means there might be some zero flows in the network. To avoid such a problem, one or more additional constraints is needed for fair bandwidth allocation. There are three possible alternatives.

- *Guaranteed allocation:* First, we assign a maximum guaranteed bandwidth $B > 0$ to all flows. Next, the rest of the bandwidth is allocated such that the maximum network throughput is achieved.

$$x^k = \sum_{p \in P^k} x^{kp} \geq B \qquad \forall\, k \in K \tag{83}$$

Let C_{max} be the maximum link capacity in the network. Then

$$C_{max} = \max_{(i,j)\in E} C_{ij} \tag{84}$$

It is always true that the traffic flow x^k of any $k \in K$ will be less than $|P^k|C_{max}$, where $|P^k|$ is the number of paths of traffic flow k. Then, the total throughput is

$$\sum_{k\in K} x^k \leq |K||P^k|C_{max} \tag{85}$$

where $|K|$ is the number of S-D pairs. Equation (83) represents the first goal of achieving the maximum guaranteed bandwidth with a target value of B. Equation (85) is the second goal of having the maximum throughput with a target value of $|K||P^k|C_{max}$. We now introduce two positive deficiency variables u and v to represent the deviation from the first goal and the second goal, respectively. Then, our two goals can be transformed into two soft constraints as

$$\sum_{p\in P^k} x^{kp} + u \geq B \tag{86}$$

and

$$\sum_{k\in K}\sum_{p\in P^k} x^{kp} + v \geq |K||P^k|C_{max} \tag{87}$$

It is clear that we want to minimize the deviations from our goals. Thus, the guaranteed allocation can be formulated as follows.

$$\text{minimize} \qquad Mu + v \tag{88}$$

such that

$$\sum_{k\in K}\sum_{p\in P^k} a_{ij}^{kp} x^{kp} \;\leq\; C_{ij} \qquad \forall\,(i,j) \in E \tag{89}$$

$$\sum_{p\in P^k} x^{kp} + u \;\geq\; B \qquad \forall\, k \in K \tag{90}$$

$$\sum_{k\in K}\sum_{p\in P^k} x^{kp} + v \;\geq\; |K||P^k|C_{max} \tag{91}$$

$$x^{kp} \;\geq\; 0 \qquad\qquad \forall\, p \in P^k, \forall\, k \in K \tag{92}$$

$$u, v \;\geq\; 0 \tag{93}$$

where M is a large positive number to give the first goal higher priority.

The maximum guaranteed bandwidth $B > 0$ is formulated as follows.

$$\text{maximize} \qquad B \tag{94}$$

such that

$$\sum_{k\in K}\sum_{p\in P^k} a_{ij}^{kp} x^{kp} \;\leq\; C_{ij} \qquad \forall\, (i,j) \in E \tag{95}$$

$$\sum_{p\in P^k} x^{kp} \;\geq\; B \qquad \forall\, k \in K \tag{96}$$

$$x^{kp} \;\geq\; 0 \qquad \forall\, p \in P^k, \forall\, k \in K \tag{97}$$

- *Mean allocation:* We start with the results obtained from the guaranteed allocation: the maximum guaranteed bandwidth B and the average bandwidth B_{avg} from the guaranteed allocation. These parameters are defined as

$$B = \min_{k\in K} \sum_{p\in P^k} x^{kp} \tag{98}$$

and

$$B_{avg} = \frac{1}{|K|} \sum_{k\in K}\sum_{p\in P^k} x^{kp} \tag{99}$$

We will divide the flows into two separate sets: K_L and K_H. K_L is the set of flows that have bandwidth less than B_{avg} and K_H is the set of flows that have bandwidth more than B_{avg}. In the first step, we will increase the traffic flows $k \in K_L$. This increment comes at the expenses of bandwidth reduction of traffic flows $k \in K_H$. The first phase is formulated as follows.

$$\text{minimize} \qquad \sum_{k \in K} (\sum_{p \in P^k} x^{kp} - B_{avg})^2 \tag{100}$$

such that

$$\sum_{k \in K} \sum_{p \in P^k} a_{ij}^{kp} x^{kp} \leq C_{ij} \qquad \forall\, (i,j) \in E \tag{101}$$

$$\sum_{p \in P^k} x^{kp} \geq B \qquad \forall\, k \in K \tag{102}$$

$$x^{kp} \geq 0 \qquad \forall\, p \in P^k, \forall\, k \in K \tag{103}$$

Since minimizing the objective function (100) prohibits some flows from bandwidth larger than B_{avg}, the next step is to assign the leftover capacity to these flows such that the total throughput is maximized. Let D^k be equal to x^k as obtained from the first phase. The second phase is formulated as follows.

$$\text{maximize} \qquad \sum_{k \in K} \sum_{p \in P^k} x^{kp} \tag{104}$$

such that

$$\sum_{k \in K} \sum_{p \in P^k} a_{ij}^{kp} x^{kp} \leq C_{ij} \qquad \forall\, (i,j) \in E \tag{105}$$

$$\sum_{p \in P^k} x^{kp} \geq D^k \qquad \forall\, k \in K \tag{106}$$

$$x^{kp} \geq 0 \qquad \forall\, p \in P^k, \forall\, k \in K \tag{107}$$

- *Proportional allocation:* We will assign each flow $k \in K$ the guaranteed bandwidth porportional to its maxflow D^k. The maxflow is the maximum throughput of the flow when there are no other flows in the network. Thus, this maxflow represents the upper bound of throughput of this flow. Our first step is to find the maxflow D^k for every $k \in K$. The maxflow problem is formulated as

$$\text{maximize} \qquad \sum_{k \in K} \sum_{p \in P^k} x^{kp} \tag{108}$$

such that

$$\sum_{p \in P^k} a_{ij}^{kp} x^{kp} \leq C_{ij} \qquad \forall\, (i,j) \in E, \forall\, k \in K \tag{109}$$

$$x^{kp} \geq 0 \qquad \forall\, p \in P^k, \forall\, k \in K \tag{110}$$

$$\tag{111}$$

In the next step, we will find the maximum guaranteed bandwidth for all flows proportional to their maxflow D^k. A traffic flow $k \in K$ is assigned a bandwidth of $\alpha^k D^k$ where $\alpha^k \in [0,1]$ is a proportional coefficient indicating a portion of its maxflow that is assigned to a flow. The goal of this step is to maximize the minimum of α^k. We define α as

$$\alpha = \min_{k \in K} \alpha^k \qquad \text{then} \qquad \alpha^k \geq \alpha \qquad \forall\, k \in K \tag{112}$$

The problem of finding the maximum α is as follows.

$$\text{maximize} \qquad \alpha \tag{113}$$

such that

$$\sum_{p \in P^k} x^{kp} = \alpha D^k \qquad \forall\, k \in K \tag{114}$$

$$\sum_{k \in K} \sum_{p \in P^k} a_{ij}^{kp} x^{kp} \leq C_{ij} \qquad \forall\, (i,j) \in E \tag{115}$$

$$x^{kp} \geq 0 \qquad \forall\, p \in P^k, \forall\, k \in K \tag{116}$$

$$\alpha \geq 0 \tag{117}$$

During the last step, the leftover bandwidth is assigned to every nonsaturated flow. This problem can be formulated as follows.

$$\text{maximize} \qquad \sum_{k \in K} \alpha^k D^k \tag{118}$$

such that

$$\sum_{p \in P^k} x^{kp} = \alpha^k D^k \qquad \forall\, k \in K \tag{119}$$

$$\sum_{k \in K} \sum_{p \in P^k} a_{ij}^{kp} x^{kp} \leq C_{ij} \qquad \forall\, (i,j) \in E \tag{120}$$

$$x^{kp} \geq 0 \qquad \forall\, p \in P^k, \forall\, k \in K \tag{121}$$

$$\alpha^k \geq \alpha \qquad \forall\, k \in K \tag{122}$$

3.4.2 Link-Based Formulation

In path-based formulation, the traffic flow will traverse only on the predertermined paths obtained from the path calculation module. Specifically, the routing matrix a_{ij}^k in single path case and a_{ij}^{kp} in multipath case obtained from the path calculation process are the inputs to the rate calculation process. These matrixes restrict other possible paths not on them.

To take *all* possible paths into account, the link-based formulation is used. The link-based formulation combines the path and rate calculations into one step. Thus, the outputs from the link-based approach are the optimal paths and their associated optimal rates.

For every S-D pair k, an artificial link (d^k, s^k) with the link cost of -1 and infinite capacity is added to the network. This special link (d^k, s^k) can only bear a single commodity k. Let $\acute{E}$ be the set of these artificial links. All costs on other links are

set to a very small positive number γ, where $0 < \gamma \ll 1$ to ensure that there is no flow circulation in the network.

In *guaranteed allocation*, the goals are to assign a maximum guaranteed bandwidth $B > 0$ to every flow and to allocate the rest of the bandwidth such that the maximum throughput is achieved. We must first determine the maximum guaranteed bandwidth B. Then, the traffic flow f^k of S-D pair k is

$$f^k = \sum_{(i,j)\in \acute{E}} f_{ij}^k \geq B \tag{123}$$

Let C_{max} be the maximum link capacity in the network. Then

$$C_{max} = \max_{(i,j)\in E} C_{ij} \tag{124}$$

It is always true that the traffic flow f^k of any $k \in K$ will be less than $T \times C_{max}$ where T is the maximum node degree. Then, the total throughput is

$$\sum_{k\in K} f^k = \sum_{k\in K} \sum_{(i,j)\in \acute{E}} f_{ij}^k \leq T|K|C_{max} \tag{125}$$

where $|K|$ is the number of S-D pairs. Equation (123) represents the first goal of achieving the maximum guaranteed bandwidth with a target value of B. Equation (125) is the second goal of having the maximum throughput with a target value of $T|K|C_{max}$. We now introduce two positive deficiency variables u and v to represent the deviation from the first goal and the second goal, respectively. Then, our two goals can be transformed into two soft constraints as

$$\sum_{(i,j)\in \acute{E}} f_{ij}^k + u \geq B \tag{126}$$

and

$$\sum_{k\in K} \sum_{(i,j)\in \acute{E}} f_{ij}^k + v \geq T|K|C_{max} \tag{127}$$

It is clear that we want to minimize the deviations from our goals. Thus, the guaranteed allocation can be formulated as follows.

minimize $$Mu + v \tag{128}$$

such that

$$\sum_{j:(i,j)\in E\cup\acute{E}} f_{ij}^k - \sum_{j:(j,i)\in E\cup\acute{E}} f_{ji}^k = 0 \qquad \forall\, i \in V, \forall\, k \in K \tag{129}$$

$$\sum_{k\in K} f_{ij}^k \le C_{ij} \qquad \forall\, (i,j) \in E \cup \acute{E} \tag{130}$$

$$\sum_{(i,j)\in\acute{E}} f_{ij}^k + u \ge B \qquad \forall\, k \in K \tag{131}$$

$$\sum_{k\in K} \sum_{(i,j)\in\acute{E}} f_{ij}^k + v \ge T|K|C_{max} \tag{132}$$

$$f_{ij}^k \ge 0 \qquad \forall\, (i,j) \in E \cup \acute{E}, \forall\, k \in K \tag{133}$$

$$u, v \ge 0 \tag{134}$$

where M is a large positive number to give the first goal higher priority.

The maximum guaranteed bandwidth $B > 0$ is formulated as follows.

maximize $$B \tag{135}$$

such that

$$\sum_{j:(i,j)\in E\cup\acute{E}} f_{ij}^k - \sum_{j:(j,i)\in E\cup\acute{E}} f_{ji}^k = 0 \qquad \forall\, i \in V, \forall\, k \in K \tag{136}$$

$$\sum_{k\in K} f_{ij}^k \le C_{ij} \qquad \forall\, (i,j) \in E \cup \acute{E} \tag{137}$$

$$\sum_{(i,j)\in\acute{E}} f_{ij}^k \ge B \qquad \forall\, k \in K \tag{138}$$

$$f_{ij}^k \ge 0 \qquad \forall\, (i,j) \in E \cup \acute{E}, \forall\, k \in K \tag{139}$$

In *mean allocation*, we start with the results obtained from the guaranteed allocation: the maximum guaranteed bandwidth B and the average bandwidth B_{avg} from the guaranteed allocation. These parameters are defined as

$$B = \min_{k \in K} f^k \tag{140}$$

and

$$B_{avg} = \frac{1}{|K|} \sum_{k \in K} f^k \tag{141}$$

We will divide the flows into two separate sets: K_L and K_H. K_L is the set of flows that have bandwidth less than B_{avg} and K_H is the set of flows that have bandwidth more than B_{avg}. In the first step, we will increase the traffic flows $k \in K_L$. This increment comes at the expenses of bandwidth reduction of traffic flows $k \in K_H$. The first phase is formulated as follows.

$$\text{minimize} \qquad \sum_{k \in K} (\sum_{(i,j) \in \acute{E}} f_{ij}^k - B_{avg})^2 \tag{142}$$

such that

$$\sum_{j:(i,j) \in E \cup \acute{E}} f_{ij}^k - \sum_{j:(j,i) \in E \cup \acute{E}} f_{ji}^k = 0 \qquad \forall\, i \in V, \forall\, k \in K \tag{143}$$

$$\sum_{k \in K} f_{ij}^k \leq C_{ij} \qquad \forall\, (i,j) \in E \cup \acute{E} \tag{144}$$

$$\sum_{(i,j) \in \acute{E}} f_{ij}^k \geq B \qquad \forall\, k \in K \tag{145}$$

$$f_{ij}^k \geq 0 \qquad \forall\, (i,j) \in E \cup \acute{E}, \forall\, k \in K \tag{146}$$

Since minimizing the objective function (142) prohibits some flows from bandwidth larger than B_{avg}, the next step is to assign the leftover capacity to these flows such that the total throughput is maximized. Let D^k be equal to f^k as obtained from the first phase. The second phase is formulated as follows.

maximize
$$\sum_{k\in K}\sum_{(i,j)\in \acute{E}} f_{ij}^k \tag{147}$$

such that

$$\sum_{j:(i,j)\in E\cup\acute{E}} f_{ij}^k - \sum_{j:(j,i)\in E\cup\acute{E}} f_{ji}^k = 0 \qquad \forall\, i \in V, \forall\, k \in K \tag{148}$$

$$\sum_{k\in K} f_{ij}^k \le C_{ij} \qquad \forall\, (i,j) \in E \cup \acute{E} \tag{149}$$

$$\sum_{(i,j)\in \acute{E}} f_{ij}^k \ge D^k \qquad \forall\, k \in K \tag{150}$$

$$f_{ij}^k \ge 0 \qquad \forall\, (i,j) \in E \cup \acute{E}, \forall\, k \in K \tag{151}$$

In *proportional allocation*, we will assign each flow $k \in K$ the guaranteed bandwidth porportional to its maxflow D^k. The maxflow is the maximum throughput of the flow when there are no other flows in the network. Thus, this maxflow represents the upper bound of throughput of this flow. Our first step is to find the maxflow D^k for every $k \in K$. The maxflow problem is formulated as

maximize
$$\sum_{k\in K}\sum_{(i,j)\in \acute{E}} f_{ij}^k \tag{152}$$

such that

$$\sum_{j:(i,j)\in E\cup\acute{E}} f_{ij}^k - \sum_{j:(j,i)\in E\cup\acute{E}} f_{ji}^k = 0 \qquad \forall\, i \in V, \forall\, k \in K \tag{153}$$

$$f_{ij}^k \le C_{ij} \qquad \forall\, (i,j) \in E \cup \acute{E}, \forall\, k \in K \tag{154}$$

$$f_{ij}^k \ge 0 \qquad \forall\, (i,j) \in E \cup \acute{E}, \forall\, k \in K \tag{155}$$

In the next step, we will find the maximum guaranteed bandwidth for all flows proportional to their maxflow D^k. A traffic flow $k \in K$ is assigned a bandwidth of

$\alpha^k D^k$ where $\alpha^k \in [0,1]$ is a proportional coefficient indicating a portion of its maxflow that is assigned to a flow. The goal of this step is to maximize the minimum of α^k. We define α as

$$\alpha = \min_{k \in K} \alpha^k \qquad \text{then} \qquad \alpha^k \geq \alpha \qquad \forall\, k \in K \tag{156}$$

The problem of finding the maximum α is as follows.

$$\text{maximize} \qquad \alpha \tag{157}$$

such that

$$\sum_{j:(i,j)\in E} f_{ij}^k - \sum_{j:(j,i)\in E} f_{ji}^k = \begin{cases} +\alpha D^k & \text{if} \quad i = s^k \\ -\alpha D^k & \text{if} \quad i = d^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i \in V, \forall\, k \in K \tag{158}$$

$$\sum_{k \in K} f_{ij}^k \leq C_{ij} \qquad \forall\, (i,j) \in E \tag{159}$$

$$f_{ij}^k \geq 0 \qquad \forall\, (i,j) \in E, \forall\, k \in K \tag{160}$$

$$\alpha \geq 0 \tag{161}$$

During the last step, the leftover bandwidth is assigned to every nonsaturated flow. This problem can be formulated as follows.

$$\text{maximize} \qquad \sum_{k \in K} \alpha^k D^k \tag{162}$$

such that

$$\sum_{j:(i,j)\in E} f_{ij}^k - \sum_{j:(j,i)\in E} f_{ji}^k = \begin{cases} +\alpha^k D^k & \text{if} \quad i = s^k \\ -\alpha^k D^k & \text{if} \quad i = d^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i \in V, \forall\, k \in K \tag{163}$$

$$\sum_{k\in K} f_{ij}^k \leq C_{ij} \qquad \forall\, (i,j) \in E \tag{164}$$

$$f_{ij}^k \geq 0 \qquad \forall\, (i,j) \in E, \forall\, k \in K \tag{165}$$

$$\alpha^k \geq \alpha \qquad \forall\, k \in K \tag{166}$$

CHAPTER IV

PERFORMANCE EVALUATION

4.1 Overview

In this chapter we present the results of the proposed framework. The results are obtained by simulation using the ns-2 network simulator on a Linux workstation. We compare our new framework with the earlier framework proposed by Koehler [24] and the traditional shortest-path algorithm. The first three algorithms are algorithms of our new proposed framework: guaranteed allocation (GA), mean allocation (MA), and proportional allocation (PA). The previous best-effort framework proposed by Koehler has two methods to select a path set: the k-shortest and the k-disjoint path set, and two methods for calculating allocation rates and fractions: the max-min, and the optimal allocation. We will use the number of paths $k = 2, 3, 4$ in Koehler's framework. The last technique considered in our comparisons is the shortest-path first (SPF). In summary, there are 16 different algorithms for which results are obtained. These algorithms are summarized in Table 1.

4.2 Network Topologies

The general procedure to obtain the results is to take a network topology and apply the traffic engineering algorithm. The results are then obtained accordingly. In this thesis, we consider five different types of network topologies. There are four network topologies generated by using the Georgia Tech Internetworking Topology Model [53] and one real IP backbone from Sprint [24].

Table 1: Framework Algorithms.

Algorithm	**Path type**	**Rate type**	**Number of paths**
SPF	shortest	N/A	1
SMM-2	shortest	max-min	2
SOP-2	shortest	optimal	2
DMM-2	disjoint	max-min	2
DOP-2	disjoint	optimal	2
SMM-3	shortest	max-min	3
SOP-3	shortest	optimal	3
DMM-3	disjoint	max-min	3
DOP-3	disjoint	optimal	3
SMM-4	shortest	max-min	4
SOP-4	shortest	optimal	4
DMM-4	disjoint	max-min	4
DOP-4	disjoint	optimal	4
GA	both	guaranteed	variable
MA	both	mean	variable
PA	both	proportional	variable

Table 2: Random Topology Parameters.

Topology	**Nodes**	**Links**	$\boldsymbol{\alpha}$
R50-Sparse	50	70	0.06
R50-Dense	50	107	0.08

4.2.1 50-Node Random Networks

Our first network topology is a flat (i.e., non-hierarchical) network structure generated by using the Georgia Tech Internetworking Topology Model [53]. Figure 3 shows an example of random network topology. This type of network does not reflect the reality of network topologies used in practice but it provides simplicity and is widely used to study networking problems. This network model places the nodes at random locations on a given plane. The links between every pair of nodes are then generated with a probability α. Therefore, this type of network requires two parameters: the number of nodes n, and the link probability α. Let m be the number of links, and β be the average node degree. We have

$$\beta = \frac{2m}{n} \tag{167}$$

and also

$$\beta = \alpha(n-1) \tag{168}$$

The number of links is given by

$$m = \alpha\frac{n(n-1)}{2} \tag{169}$$

We will consider two types of random networks: sparse and dense topologies. In the sparse random topology, we assign a link probability of 0.06. The dense random topology has a link probability of 0.08. Table 2 summarizes the parameters used to construct the topologies.

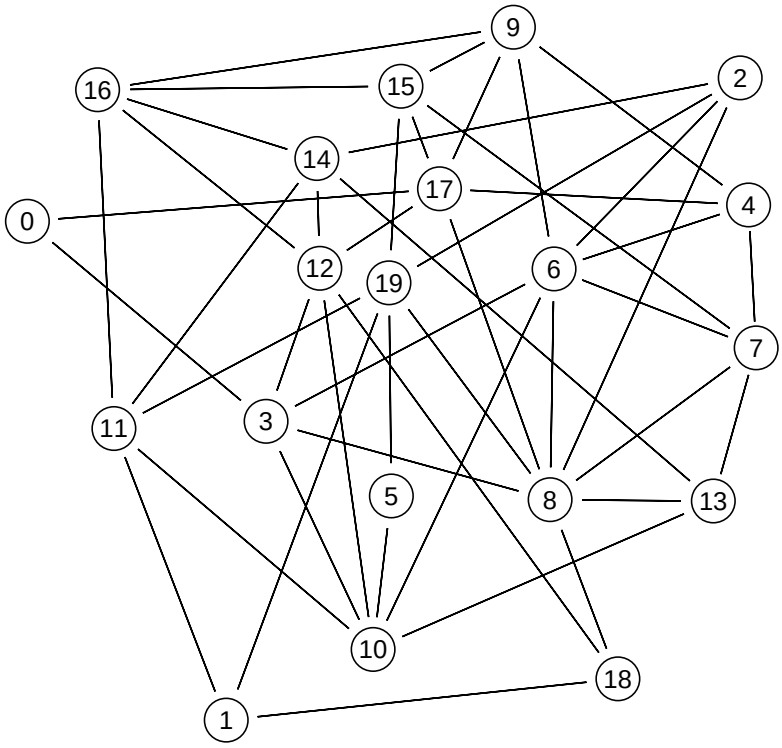

Figure 3: Random Network Topology.

4.2.2 50-Node Transit-Stub Networks

Our next network topology is a transit-stub network generated by using the Georgia Tech Internetworking Topology Model [53]. This type of network represents a hierarchical structure existing in a real network. Stub domains correspond to the customer networks connected to the backbone networks represented by transmit domains. The model constructs the network as follows. First, the random topology is created with each node representing an entire transit domain. Next, for each node in that topology, we generate a random topology and replace every node in the former topology with the newly generated topology. The newly generated topologies represent the transit domains. Next, for every node in a transit domain, we create multiple random topologies representing stub domains. We replace every node in a transit domain with the created stub domains. Last, we add some extra links between pairs of nodes. The example of transit-stub network is shown in Figure 4.

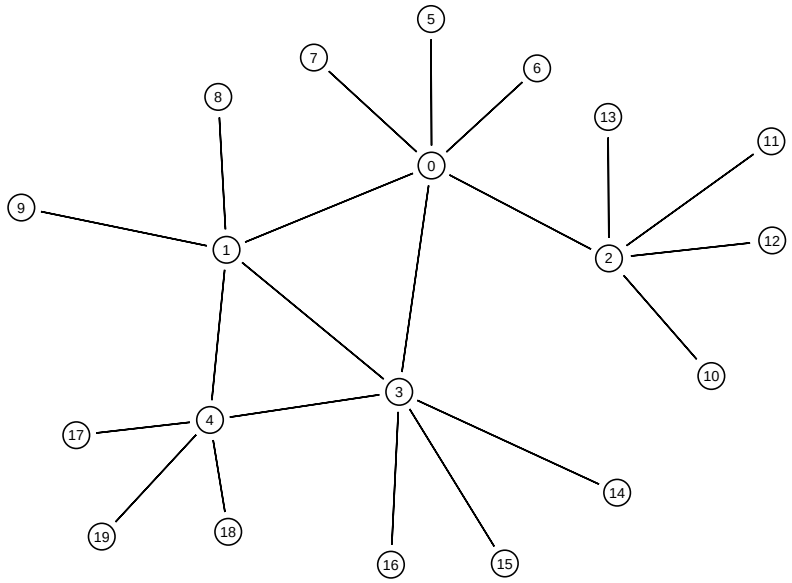

Figure 4: Transit-Stub Network Topology.

The transit-stub topology is controlled by the following parameters

$$T = \text{number of transit domains} \tag{170}$$

$$S = \text{number of stub domains per transit node} \tag{171}$$

$$N_t = \text{average number of nodes per transit domain} \tag{172}$$

$$N_s = \text{average number of nodes per stub domain} \tag{173}$$

$$\alpha_t = \text{link probability within a transit domain} \tag{174}$$

$$\alpha_s = \text{link probability within a stub domain} \tag{175}$$

The total number of nodes n is given by

$$n = T \times N_t \times (T + S \times N_s) \tag{176}$$

We study two types of transit-stub networks: sparse and dense topologies. Table 3 summarizes the parameters used to construct the two transit-stub topologies.

Table 3: Transit-Stub Topology Parameters.

Topology	Nodes	Links	T	S	n_t	n_s	α_t	α_s
TS50-Sparse	50	64	1	1	5	9	0.5	0.25
TS50-Dense	50	114	1	1	5	9	0.9	0.5

4.2.3 Sprint IP Backbone

A generic high level Internet Service Provider (ISP) network consists of multiple Point of Presences (POPs) interconnected via backbone network. A POP is a physical place where multiple routers are located. Each POP has some Access Routers (ARs) that are the entry points for customers to the ISP network. AR aggregates many low-speed interfaces connecting to customers into a few high-speed interfaces connecting to the backbone network. Thus, POP is regarded as a traffic aggregator where flows from many users are combined. Also inside each POP, some Backbone Routers (BRs) are connected via Wide Area Network (WAN) lines to other POPs to form the backbone network.

Unfortunately, real ISP network topology is not publicly available because most ISPs regard their router-level configuration as classified information. In this thesis, we limit our model of the backbone network at the POP level where a POP represents a network node where traffic aggregates. Even though this high level view of the Sprint network topology is not complete, we believe that it will provide a more realistic network topology than the ones generated from GT-ITM [53].

4.3 Evaluation Methodology

We compare our new framework with the earlier framework proposed by Koehler [24] and the traditional shortest-path algorithm. There are 16 algorithms considered in this research as summarized in Table 1. All algorithms are compared on the five different topologies discussed in section 4.2.

4.3.1 Procedures

The general procedures to obtain the results are summarized as follows.

1. Generate a network topology $G(V, E, C)$. The network topology is either R50-Sparse, R50-Dense, TS50-Sparse, TS50-Dense, or Sprint topology.

2. Select a random subset of source-destination pairs $k \in K$.

3. Apply the traffic engineering algorithm to obtain the path set and its associated rate.

   - *SPF algorithm:*
     - *Number of paths:* The number of paths used for each S-D pair is equal to one.
     - *Path set:* The path set P^k for each S-D pair $k \in K$ is calculated using the shortest-path routing algorithm.
     - *Path rates:* The associated rate is not computed because in SPF algorithm each path does not have explicit rate.
   - *Koehler's framework algorithms:* SOP, SMM, DOP, DMM
     - *Number of paths:* The number of paths used for each S-D pair is either 2,3, or 4 paths.
     - *Path set:* The path set P^k for each S-D pair $k \in K$ is either k-shortest or k-disjoint depending on the algorithm.
     - *Path rates:* For the selected path set, the associated rate x_p is determined by either max-min allocation or optimal allocation depending on the algorithm.
   - *New framework algorithms:* GA, MA, PA
     - *Number of paths:* The number of paths used for each S-D pair is not necesseary specified.

- *Path set and rates:* The path set P^k for each S-D pair $k \in K$ and the associated rate x^{kp} are obtained directly from the optimization.

4. Calculate the fractional rate ω_{ij}^k for every link $(i,j) \in E$, and every S-D pair $k \in K$. A flow rate f_{ij}^k on link (i,j) of S-D pair $k \in K$ can be determined from the path flow x^{kp}. The fractional rate ω_{ij}^k is given by

$$\omega_{ij}^k = \frac{f_{ij}^k}{\sum_{k \in K} f_{ij}^k} \tag{177}$$

Note that

$$\sum_{k \in K} \omega_{ij}^k = 1 \tag{178}$$

5. Set the fractional rate ω_{ij}^k as the weight setting on the queue entity at node $i \in E$.

6. Calculate the traffic splitting ratio ϕ^{kp} of path $p \in P^k$ so that the source node of the S-D pair $k \in K$ can distribute its traffic proportionally over multiple paths. The traffic splitting ratio ϕ_p is given by

$$\phi^{kp} = \frac{x^{kp}}{\sum_{p \in P^k} x^{kp}} \tag{179}$$

Note that

$$\sum_{p \in P^k} \phi^{kp} = 1 \tag{180}$$

7. Generate multiple TCP flows for every S-D pair.

8. Collect flow statistics.

4.3.2 Metrics

The goals of this research are to deliver more equal shares of bandwidth to best-effort users as compared to the traditional shortest-path routing algorithm and to maximize the total allocated bandwidth to users. We will use two metrics to evaluate our dual objectives.

Max-min fairness is the most commonly used fairness criterion in the networking literature. An allocation of bandwidths is max-min fair if it is not possible to increase the assigned bandwidth of any flow at the expenses of the assigned bandwidth of flows that have larger shares. It turns out that the max-min fairness is equivalent to another fairness definition that the list of allocated bandwidth, when sorted in non-decreasing order, is lexicographically maximum [22].

Definition 1 *A vector X is a n-dimensional vector whose coordinates are sorted in non-decreasing order, i.e.,*

$$x_1 \leq x_2 \leq \cdots \leq x_n$$

is lexicographically greater than another vector Y whose coordinates are also sorted in non-decreasing order, i.e.,

$$y_1 \leq y_2 \leq \cdots \leq y_n$$

if there exists a number s, $1 \leq s \leq n$ such that $x_i = y_i$ for $i = 1, 2, \ldots s-1$ and $x_s > y_s$.

For example, $x = (2, 3, 4)$ is lexicographically greater than $y = (1, 4, 5)$. It is clear from the above definition that if $x_1 = \min x_i$ is larger than $y_1 = \min y_i$, then X is lexicographically greater than Y. In other words, X is fairer than Y in the max-min sense.

The metrics that we use for performance evaluation of algorithms in this research are the minimum received bandwidth Bw_{min} and the average received bandwidth Bw_{avg}. The minimum received bandwidth Bw_{min} measures how well the algorithm allocates bandwidth to the least happiest users. This is a metric to measure fairness based on lexicographically maximum definition. The second metric is the average received bandwidth Bw_{avg}. This is a network-oriented performance metric. It quantifies how well the algorithm utilizes the total network bandwidth.

We define the minimum received bandwidth Bw_{min} and the average received bandwidth Bw_{avg} as

$$Bw_{min} = \min_{k \in K} Bw_k \tag{181}$$

and

$$Bw_{avg} = \frac{1}{|K|} \sum_{k \in K} Bw_k \tag{182}$$

To summarize the results, we use η_{min}^{SPF} and η_{avg}^{SPF} to indicate the performance improvement over the SPF. We define

$$\eta_{min}^{SPF} = \frac{Bw_{min} - Bw_{min}^{SPF}}{Bw_{min}^{SPF}} \tag{183}$$

and

$$\eta_{avg}^{SPF} = \frac{Bw_{avg} - Bw_{avg}^{SPF}}{Bw_{avg}^{SPF}} \tag{184}$$

4.4 Framework Results

The results obtained for each five network topologies are presented in this section. The topologies considered in this research are two random topologies: R50-Sparse, and R50-Dense; two transit-stub topologies: TS50-Sparse, and TS50-Dense; and the Sprint topology. In all cases, we use the number of paths $k = 2, 3, 4$ in Koehler's framework algorithms.

4.4.1 Random Topology Results

The two random topologies are depicted in Figure 5 and 6. Both the R50-Sparse and R50-Denses have 50 nodes. The R50-Sparse consists of 70 bi-directional links, whereas the R50-Denses has 107 links. Each link is bi-directional with a capacity of 10 Mbps.

The configuration parameters used to obtain the results of the traffic engineering schemes are summarized in Table 4.

Table 5 shows the average number of paths found by each algorithm for both topologies. The results of the random topologies are shown in Table 6 and 7.

Table 4: Simulation Parameters.

Parameters	**Values**
Traffic type	TCP
Packet size	1500 bytes
Data size	50 Mbytes
Number of Flows	100 flows
Simulation time	100 seconds

4.4.1.1 Sparse Topology Results

The results of the R50-Sparse topology are shown in Table 6. The best algorithms of Koehler's framework, regarding Bw_{min}, are DOP-2, DOP-3, DOP-4, SOP-2, SOP-3, and SOP-4. They perform approximately equally well and are significantly better than the SPF, on the order of 40-50%. Relating to Bw_{avg}, all algorithms are better than the SPF. The DOP-2, DOP-3, DOP-4 have the best improvements over the SPF, on the order of 10%. The obvious conclusion is that Koehler's framework achieves the best overall results for both Bw_{min} and Bw_{avg} when the optimal rate allocation is used with the disjoint path sets.

The best algorithms of the new framework, in terms of Bw_{min}, are the GA and MA. They achieve considerable improvements over Koehler's framework and the SPF on the order of 40% and 110%, respectively. The PA is on a par with Koehler's best algorithms. From the resource usage standpoint, or Bw_{avg}, all new framework's algorithms are able to deliver better performance than Koehler's best algorithm and the SPF. The GA is the best candidate, whereas the MA and PA are just short of the best one. They gain performance improvements on Bw_{avg} over Koehler's best algorithm and the SPF, on the order of 10% and 20%, respectively.

In summary, we observe significant performance improvements of the new proposed framework over Koehler's framework and the SPF. The GA and MA are the best candidates regarding both Bw_{min} and Bw_{avg}.

4.4.1.2 Dense Topology Results

The results of the R50-Dense topology are shown in Table 7. We observe that all algorithms perform better in R50-Dense topology. With respect to Bw_{min}, the best algorithms of Koehler's framework are SOP-2, SOP-3, SOP-4, DOP-2, DOP-3, and DOP-4. They perform equally well but are only slighly better than the SPF algorithm. However, with regard to Bw_{avg}, all Koehler's algorithms perform better than the SPF. The SOP-3 and SOP-4 are the best Koehler's framework algorithms. They are almost 30% higher than the SPF. Other algorithms that employ the optimal allocation perform close to the best ones. It is clear that Koehler's framework produces best results when the optimal rate allocation is used.

The best algorithms of the new framework are again the GA and MA. They are slightly better than the SPF algorithm and are at the same level as Koehler's best algorithms in terms of Bw_{min}. However, for Bw_{avg}, the GA and MA outperform Koehler's best algorithms, the SOP-3 and SOP-4, by 18%. In comparison with the SPF, the GA and MA algorithms have performance gain on the order of 50%.

In the case of R50-Dense topology, we expect to gain considerable improvements over SPF because there are more path choices than the R50-Sparse as shown in Table 5. The results show that we do not have better improvements in Bw_{min} over the SPF as in the case of R50-Sparse. However, we obtain substantial improvements in Bw_{avg} over the SPF. One reason is that since there are more alternative paths available, it is less likely that the SPF will happen to choose the paths traversing the same link. Therefore, the hot spots in the network are avoided.

In summary, both new proposed framework and Koehler's framework are able to gain only some enhancements in Bw_{min} over the SPF. However, both frameworks provide dramatic improvements in Bw_{avg}. In this regard, the new framework algorithms, the GA and MA, are the best candidates.

Table 5: Summary of average number of paths found in R50-Sparse and R50-Dense topologies.

Algorithm	Average Paths Found	
	R50-Sparse	**R50-Dense**
SPF	1	1
2-Shortest	1.72	1.86
3-Shortest	2.32	2.63
4-Shortest	2.85	3.35
2-Disjoint	1.54	1.81
3-Disjoint	1.79	2.55
4-Disjoint	1.82	2.90
GA	1.31	2.05
MA	1.40	2.07
PA	1.35	2.24

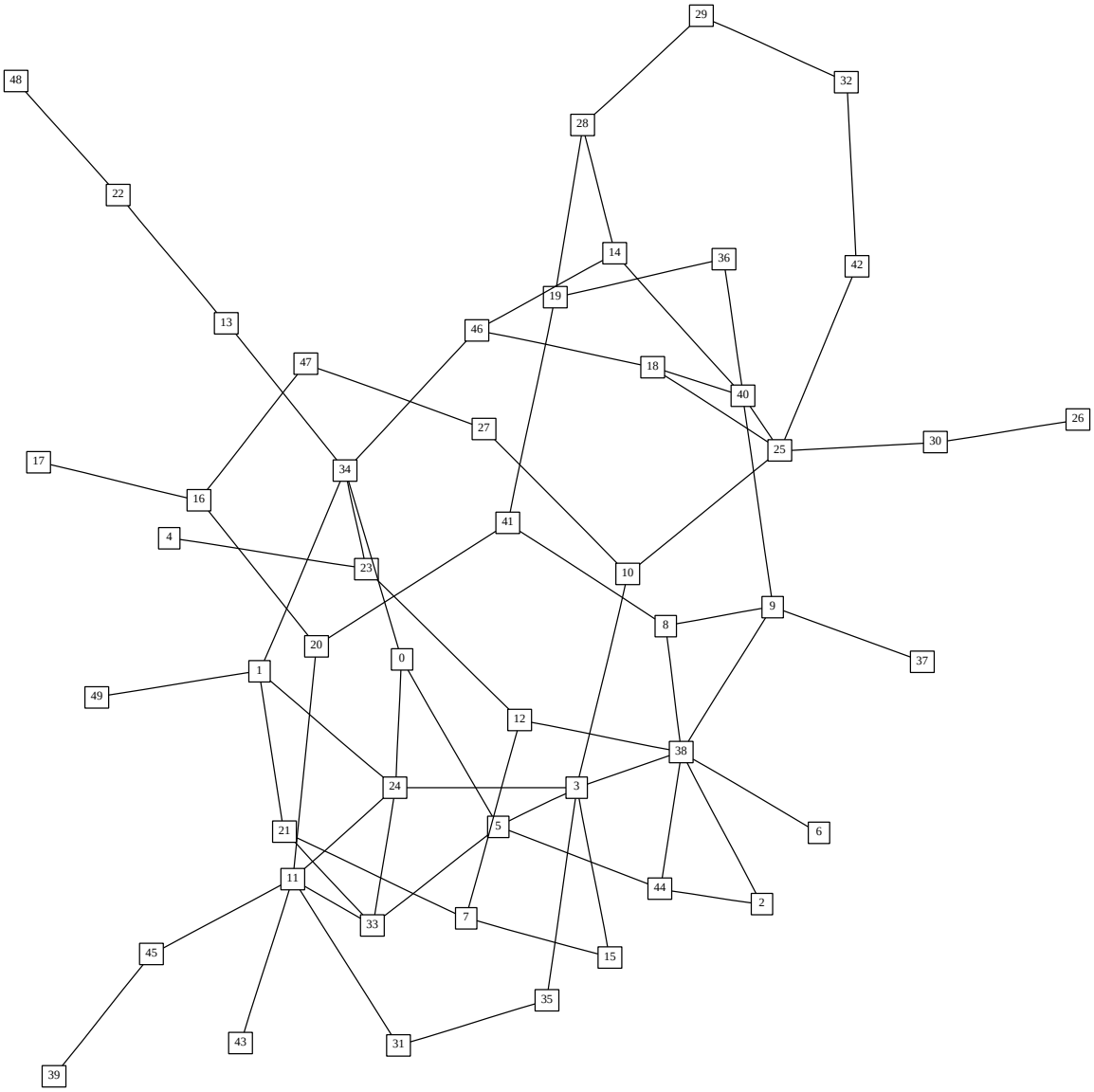

Figure 5: R50-Sparse Topology.

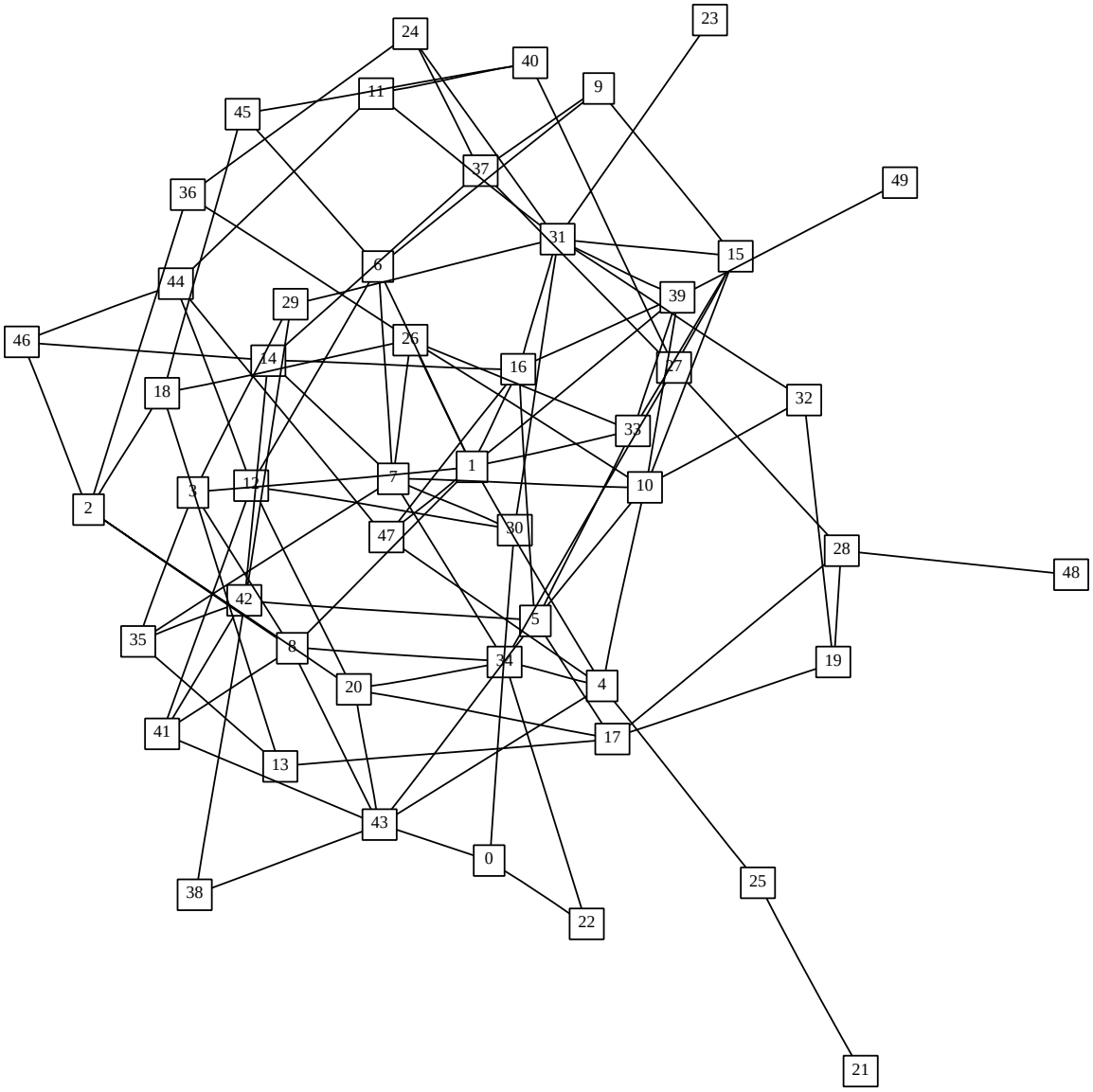

Figure 6: R50-Dense Topology.

Table 6: R50-Sparse Results.

Algorithm	$\boldsymbol{Bw_{min}}$ (Mbps)	$\boldsymbol{\eta_{min}^{SPF}}$	$\boldsymbol{Bw_{avg}}$ (Mbps)	$\boldsymbol{\eta_{avg}^{SPF}}$
SPF	0.56	-	2.71	-
SMM-2	0.36	-0.36	2.73	0.01
SMM-3	0.33	-0.41	2.73	0.01
SMM-4	0.24	-0.57	2.68	-0.01
DMM-2	0.49	-0.13	2.89	0.07
DMM-3	0.45	-0.20	2.86	0.06
DMM-4	0.46	-0.18	2.87	0.06
SOP-2	0.78	0.39	2.79	0.03
SOP-3	0.82	0.46	2.87	0.06
SOP-4	0.82	0.46	2.86	0.06
DOP-2	0.82	0.46	2.93	0.08
DOP-3	0.81	0.45	2.95	0.09
DOP-4	0.85	0.52	2.95	0.09
GA	1.18	1.11	3.33	0.23
MA	1.25	1.23	3.12	0.15
PA	0.81	0.45	3.22	0.19

Table 7: R50-Dense Results.

Algorithm	$\boldsymbol{Bw_{min}}$ **(Mbps)**	$\boldsymbol{\eta_{min}^{SPF}}$	$\boldsymbol{Bw_{avg}}$ **(Mbps)**	$\boldsymbol{\eta_{avg}^{SPF}}$
SPF	1.57	-	3.94	-
SMM-2	1.40	-0.11	4.45	0.13
SMM-3	1.05	-0.33	4.68	0.19
SMM-4	0.78	-0.50	4.57	0.16
DMM-2	0.94	-0.40	4.37	0.11
DMM-3	0.82	-0.48	4.32	0.10
DMM-4	0.72	-0.54	4.34	0.10
SOP-2	1.70	0.08	4.67	0.19
SOP-3	1.73	0.10	5.01	0.27
SOP-4	1.69	0.08	5.07	0.29
DOP-2	1.62	0.03	4.47	0.13
DOP-3	1.72	0.10	4.68	0.19
DOP-4	1.70	0.08	4.74	0.20
GA	1.66	0.06	5.96	0.51
MA	1.68	0.07	5.81	0.47
PA	1.46	-0.07	5.23	0.33

4.4.2 Transit-Stub Topology Results

We consider two transit-stub topologies. They are illustrated in Figure 7 and 8. The TS50-Sparse and TS50-Denses have 50 nodes. In the case of TS50-Sparse, the topology has 64 links. The TS50-Denses has 114 links. Each link is bi-directional with a capacity of 10 Mbps.

Table 8 shows the average number of paths found by each algorithm for both topologies. The results of the transit-stub topologies are shown in Table 9 and 10.

4.4.2.1 Sparse Topology Results

The results of the TS50-Sparse topology are shown in Table 9. We do not observe significant enhancements from Koehler's framework regarding Bw_{min} in the TS50-Sparse topology. All algorithms that apply the optimal allocation are on a par with the SPF. However, some performance improvements are possible with the new proposed framework.

With respect to Bw_{avg}, all Koehler's framework algorithms have improvements over the SPF, on the order of 10-20%. We obtain higher performance improvements with the new proposed framework. All of the new algorithms, the GA, MA and PA, yield 10-20% and 30-40% greater than Koehler's framework and the SPF, respectively.

In summary, we notice that the new proposed framework has performance gains over Koehler's framework and the SPF in both Bw_{min} and Bw_{avg}. The GA is the best candidate in both regards, while the MA and PA are short of the best one.

4.4.2.2 Dense Topology Results

The results of the TS50-Dense topology are shown in Table 10. We first analyze the results with respect to Bw_{min}. The best algorithms of Koehler's framework are DOP-2, DOP-3, and DOP-4. They outperform the SPF algorithm by over 50%. With regard to Bw_{avg}, the SMM-4 is the best Koehler's algorithm. It improves upon the SPF greater than 50%. In addition, all other Koehler's algorithms perform better

than the SPF and are not far behind from the best one. The best algorithm in both criteria is not clear but if we give the Bw_{min} a higher priority, the DOP-4 is the best one.

In both Bw_{min} and Bw_{avg} cases, all algorithms of the new proposed framework outperform both Koehler's framework and the SPF. They function equally well in both regards. In the Bw_{min} case, they attain a performance gain on the order of 10% and 60% over Koehler's framework and the SPF, respectively. For Bw_{avg}, the new framework excels Koehler's framework and the SPF by an order of 20% and 80%, respectively.

In summary, we obtain performance gains over the SPF from both new proposed framework and Koehler's framework. All algorithms of the new proposed framework perform better than the Koehler's algorithms and the SPF algorithm.

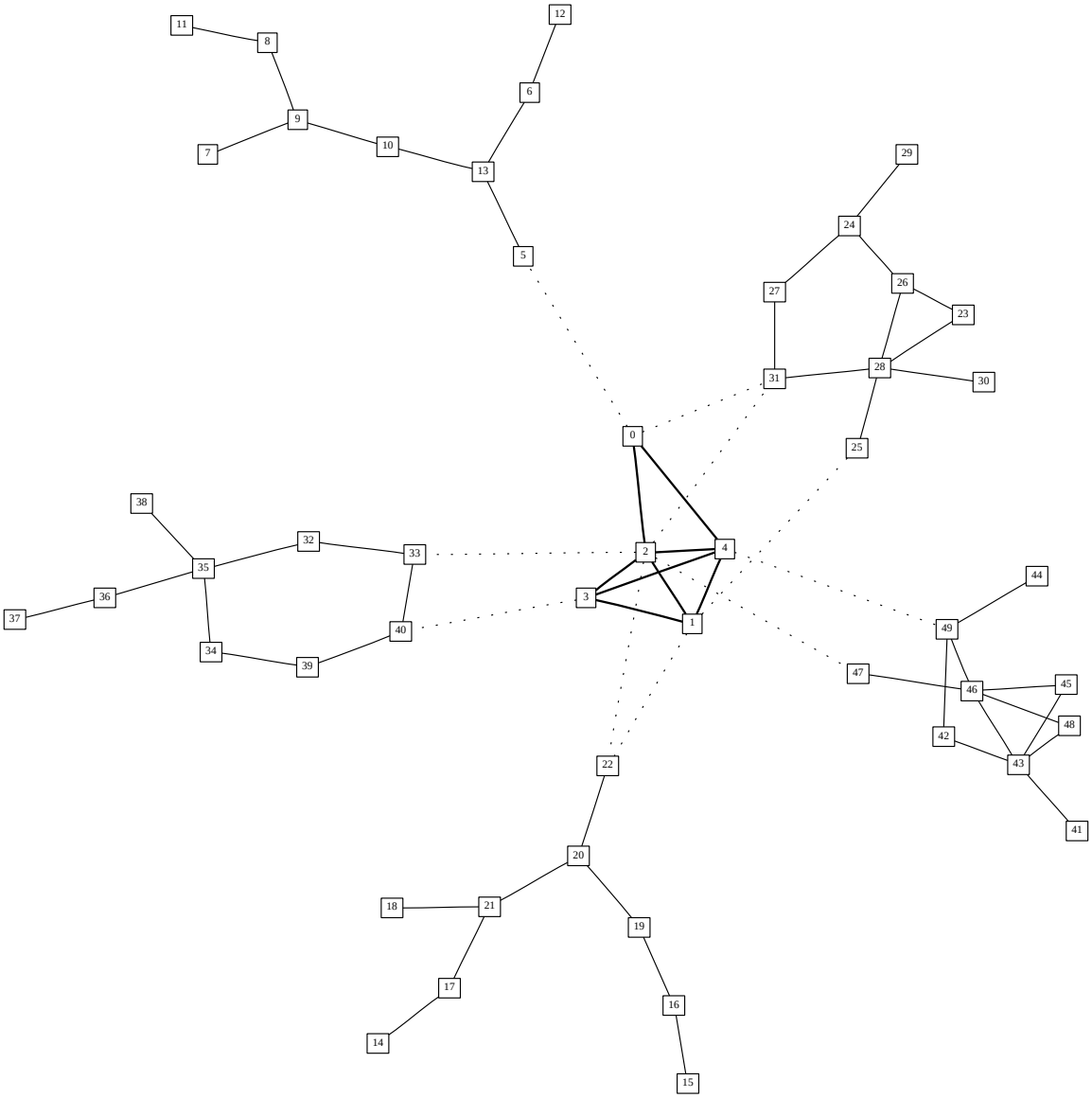

Figure 7: TS50-Sparse Topology.

Table 8: Summary of average number of paths found in TS50-Sparse and TS50-Dense topologies.

Algorithm	Average Paths Found	
	TS50-Sparse	**TS50-Dense**
SPF	1	1
2-Shortest	1.88	1.96
3-Shortest	2.64	2.90
4-Shortest	3.32	3.82
2-Disjoint	1.27	2.00
3-Disjoint	1.29	2.33
4-Disjoint	1.30	2.46
GA	1.14	1.43
MA	1.14	1.55
PA	1.14	1.48

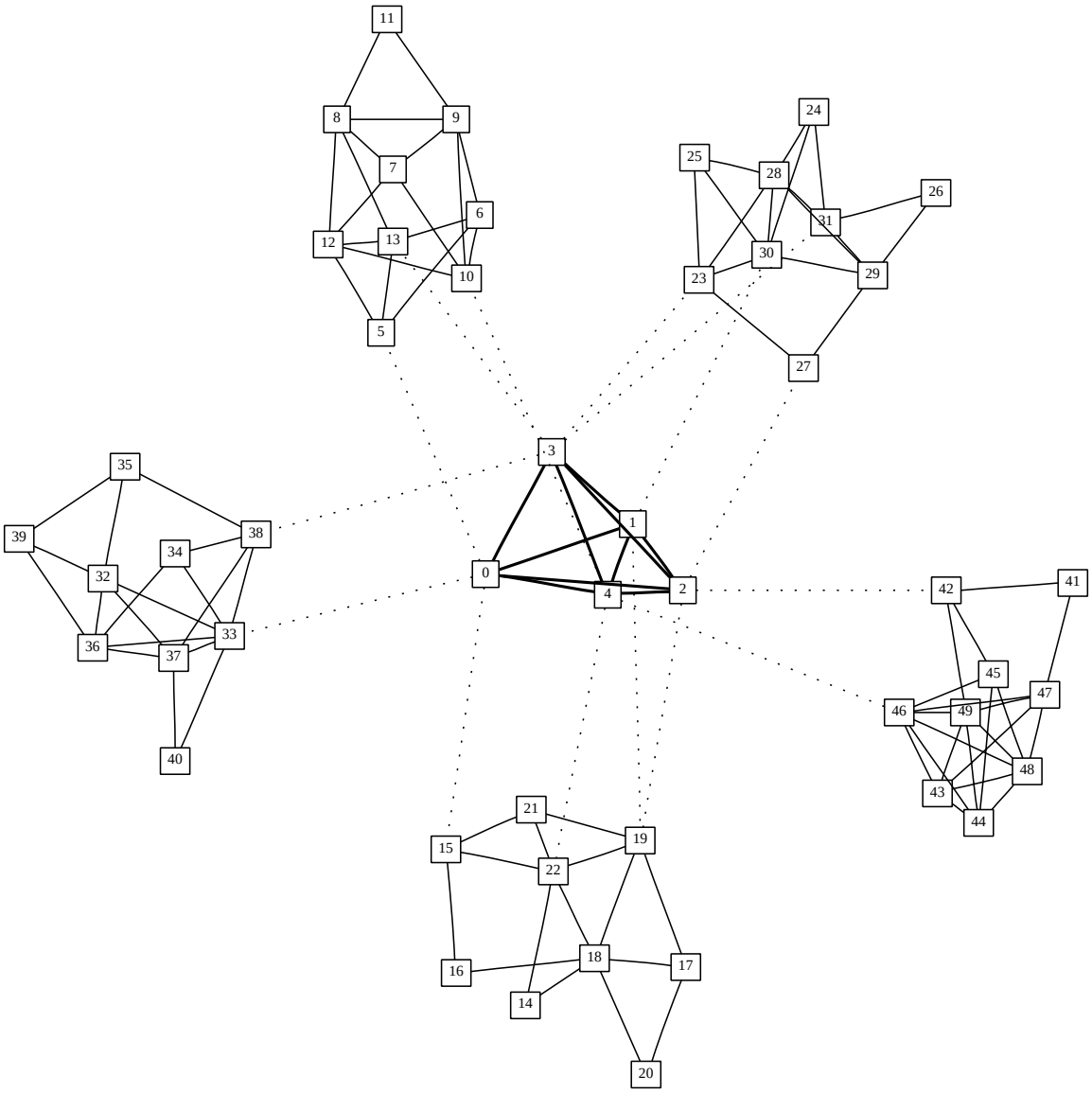

Figure 8: TS50-Dense Topology.

Table 9: TS50-Sparse Results.

Algorithm	$\boldsymbol{Bw_{min}}$ **(Mbps)**	$\boldsymbol{\eta_{min}^{SPF}}$	$\boldsymbol{Bw_{avg}}$ **(Mbps)**	$\boldsymbol{\eta_{avg}^{SPF}}$
SPF	0.36	-	1.62	-
SMM-2	0.30	-0.17	1.83	0.13
SMM-3	0.22	-0.39	1.90	0.17
SMM-4	0.20	-0.44	1.89	0.17
DMM-2	0.32	-0.11	1.86	0.15
DMM-3	0.33	-0.08	1.83	0.13
DMM-4	0.37	0.03	1.86	0.15
SOP-2	0.35	-0.03	1.78	0.10
SOP-3	0.36	0.00	1.90	0.17
SOP-4	0.36	0.00	1.89	0.17
DOP-2	0.35	-0.03	1.81	0.12
DOP-3	0.36	0.00	1.90	0.17
DOP-4	0.36	0.00	1.88	0.16
GA	0.39	0.08	2.32	0.43
MA	0.40	0.11	2.13	0.31
PA	0.38	0.06	2.21	0.36

Table 10: TS50-Dense Results.

Algorithm	$\boldsymbol{Bw_{min}}$ **(Mbps)**	$\boldsymbol{\eta_{min}^{SPF}}$	$\boldsymbol{Bw_{avg}}$ **(Mbps)**	$\boldsymbol{\eta_{avg}^{SPF}}$
SPF	0.43	-	2.46	-
SMM-2	0.55	0.28	3.16	0.28
SMM-3	0.45	0.05	3.46	0.41
SMM-4	0.25	-0.42	3.64	0.48
DMM-2	0.53	0.23	3.34	0.36
DMM-3	0.56	0.30	3.65	0.48
DMM-4	0.53	0.23	3.81	0.55
SOP-2	0.48	0.12	3.24	0.32
SOP-3	0.49	0.14	3.52	0.43
SOP-4	0.58	0.35	3.63	0.48
DOP-2	0.65	0.51	3.47	0.41
DOP-3	0.65	0.51	3.53	0.43
DOP-4	0.66	0.53	3.63	0.48
GA	0.72	0.67	4.55	0.85
MA	0.72	0.67	4.53	0.84
PA	0.71	0.65	4.50	0.83

4.4.3 ISP Topology Results

The Sprint backbone network is shown in Figure 9. It composes of 13 nodes and 27 bi-directional links. The bandwidth of every link is 45 Mbps. There are a total of 50 source-destination pairs generating traffic.

Table 11 shows the average number of paths found by each algorithm for the Sprint backbone topology.

The results in Table 12 show that the GA and MA are the best candidates. Both algorithms significantly improve the Bw_{min} over the SPF in the range of 100%. The previous framework by Koehler also gains enhancement over the SPF in the order of 60% when the disjoint path sets and optimal allocation are coupled.

In terms of Bw_{avg}, all new framework algorithms delivery a performance improvement of 20-25% over the SPF. Koehler's framework using optimal allocation has performance gain in the order of 20% over the SPF. The disjoint and shortest path sets are not distinguishable in this regard.

It is clear that the new framework algorithms, GA and MA, are overall superior. The disjoint path set with optimal allocation algorithms are the best in Koehler's framework. They deliver Bw_{min} less than the GA and MA. However, they closely match the best algorithms regarding Bw_{avg}.

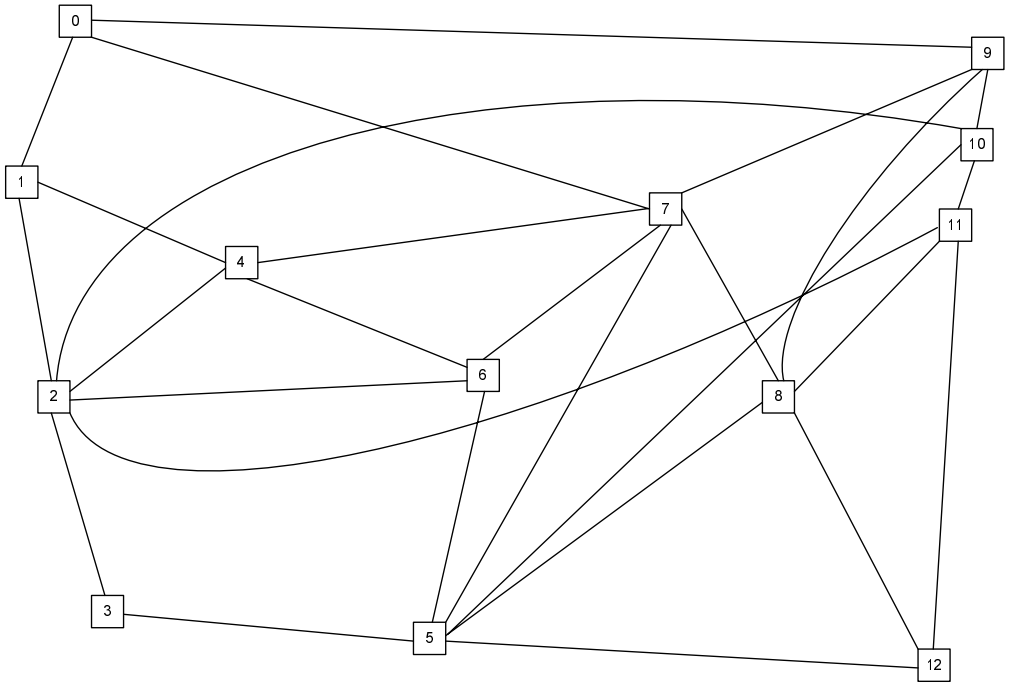

Figure 9: Sprint IP Backbone Topology.

Table 11: Summary of average number of paths found in Sprint backbone topology.

Algorithm	**Average Paths Found**
SPF	1.00
2-Shortest	1.94
3-Shortest	2.73
4-Shortest	3.46
2-Disjoint	2.00
3-Disjoint	2.85
4-Disjoint	3.31
GA	1.28
MA	1.60
PA	1.58

Table 12: Sprint Results.

Algorithm	Bw_{min} (Mbps)	η_{min}^{SPF}	Bw_{avg} (Mbps)	η_{avg}^{SPF}
SPF	7.7	-	16.8	-
SMM-2	6.9	-0.10	19.9	0.18
SMM-3	6.8	-0.12	19.6	0.17
SMM-4	8.3	0.08	18.7	0.11
DMM-2	10.3	0.34	19.8	0.18
DMM-3	7.9	0.03	18.4	0.10
DMM-4	6.1	-0.21	17.5	0.04
SOP-2	9.5	0.23	20.2	0.20
SOP-3	10.5	0.36	19.9	0.18
SOP-4	11.5	0.49	19.8	0.18
DOP-2	12.0	0.56	20.4	0.21
DOP-3	12.5	0.62	19.7	0.17
DOP-4	11.8	0.53	19.7	0.17
GA	15.1	0.96	20.8	0.24
MA	15.5	1.01	20.2	0.20
PA	9.93	0.29	20.7	0.23

CHAPTER V

CONCLUSIONS, CONTRIBUTIONS, AND FUTURE RESEARCH

5.1 Conclusions

In this thesis, we have proposed a new best-effort framework with three bandwidth allocation algorithms, GA, MA, and PA. The performance of the new framework is compared with those of the traditional shortest-path routing and the earlier framework developed by Koehler [24].

To compare the performance of the different algorithms, we used two performance metrics, Bw_{min} and Bw_{avg}. The Bw_{min} indicates how fairly the algorithm allocates the shared bandwidth in max-min sense. The second metric Bw_{avg} represents how well the algorithm utilizes the available network bandwidth. The best candidate must provide superior performance in terms of both metrics.

We have shown in the previous chapter that the new framework improves bandwidth allocation over the SPF and the earlier work by Koehler. However, the performance gain is topology dependent.

5.1.1 Random Topologies

The best candidates are the GA and MA in both sparsely and densely connected network. We have found that the new framework provides larger performance gains when the network is sparsely connected.

When the network is sparsely connected, we achieve considerable improvements on Bw_{min} over Koehler's framework best algorithm and the SPF on the order of 40% and 110%, respectively. Regarding Bw_{avg}, the new framework gains performance

improvements over Koehler's best algorithm and the SPF, on the order of 10% and 20%, respectively.

When the network is densely connected, we can significantly improve the Bw_{avg} whereas the Bw_{min} has marginal improvements over the SPF. Regarding Bw_{avg}, the new framework outperforms Koehler's best algorithms by 18%. In comparison with the SPF, the new framework has a performance gain on the order of 50%.

5.1.2 Transit-Stub Topologies

All new framework algorithms are the best candidates. They perform approximately well in both sparsely and densely connected network. It is interesting that the performance gains are in reverse order from the random topologies. We have large performance gain on both metrics in dense network, while only Bw_{avg} can be improved in the sparse network.

In the sparsely connected network, we observe 10-20% and 30-40% improvement on Bw_{avg} over Koehler's framework best algorithm and the SPF, repectively. There is no significant enhancement on Bw_{min}.

When the network is densely connected, the performance gains are noticeable on both metrics. In the Bw_{min} case, we attain a performance gain on the order of 10% and 60% over Koehler's framework best algorithm and the SPF, respectively. For Bw_{avg}, the new framework excels Koehler's framework best algorithm and the SPF by an order of 20% and 80%, respectively.

5.1.3 ISP Topology

Again, the GA and MA are the best candidates. They are better on Bw_{min} than Koehler's framework best algorithm and the SPF by 24% and 100%, respectively. In terms of Bw_{avg}, the new framework and Koehler's framework match in performance. They have improvement over the SPF, on the order of 20%.

5.2 Contributions

We developed a new best-effort traffic engineering framework with an objective to deliver better shares of bandwidth to users as compared with the current shortest-path routing. The main features of the new framework are:

- *Simple:* The calculation is needed only one time at the initialization phase. There is no extra protocol required except MPLS. The simplicity of the framework allows easy deployment.

- *Static:* The input of the framework is limited to network topology. The possibility of routing oscillation that will degrade the performance is avoided.

- *Centralized:* The central traffic engineering server performs optimization. This approach allows the switches to offload all computations onto the traffic engineering server.

- *Offline:* The paths and rates are computed beforehand.

- *Multipath:* The multiple *good* paths instead of the *best* one are used. This allows us to achieve load balancing.

- *Best-effort:* The user demand volumes are not known *a priori*.

We proposed three algorithms to calculate paths and rates based on this framework. All of these algorithms were implemented in the AMPL/CPLEX optimization software package, and incorporated into the ns-2 network simulator for performance evaluation. The MPLS extension to the ns-2 has been modified to support our proposed algorithms. Supplementary functions have been added to the ns-2 to record statistical information, and to generate traffic.

We have evaluated our new framework in five different network topologies. There are two random topologies, two transit-stub topologies, and one real ISP topology. The results of the new framework were compared with those of the traditional shortest-path routing, and the ealier work by Koehler.

5.3 Future Research

The possibilities in future research are as follow:

- It would be educative to study the new framework in a large and more detailed ISP backbone networks. We only modeled the ISP backbone network at the POP level where a POP represents a network node where traffic aggregates. This is because real ISP network is not publicly available. Most ISPs regard their router-level configuration as classified information. However, we can use network mapping tools to discover and measure real ISP networks. Further study can now proceed with the new micro detailed information.

- In this reseach we assume the best-effort traffic demand is not well defined, and not known *a priori*. By this, we argue that traffic measurement is unable to accurately quantify the real *intrinsic* best-effort demands because of their elastic nature. However, if the information of aggregated traffic distributions is available, by measurement or traffic modeling, this information can be used as additional constraints in the optimization problem. Furthermore, we can use stochastic optimization approach that normally yeilds better results than the deterministic counterpart.

- The adaptive traffic engineering scheme is worth investigating. We have known that adaptive schemes utilizing feedback between routing algorithm and flow pattern may lead to routing oscillation. However, network state information gives us a more detailed and current condition of the network. With careful

design and consideration, some state information may enhance the performance.

APPENDIX A

DETAILED RESULTS

50-Node Sparse Random Topology

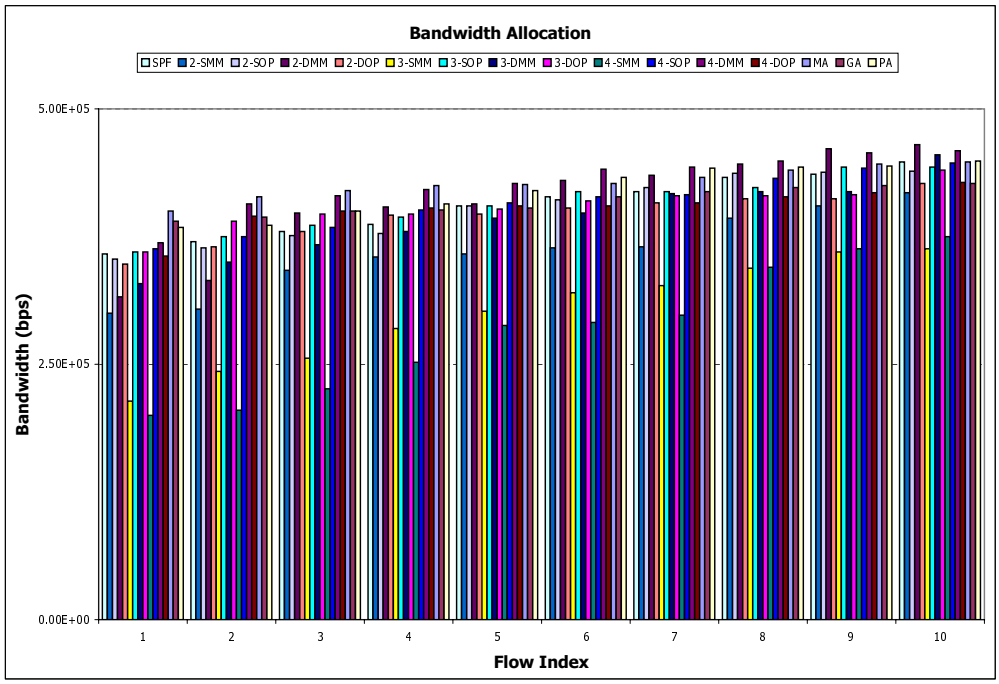

Figure 10: R50-Sparse - Bandwidth of flows 1–10.

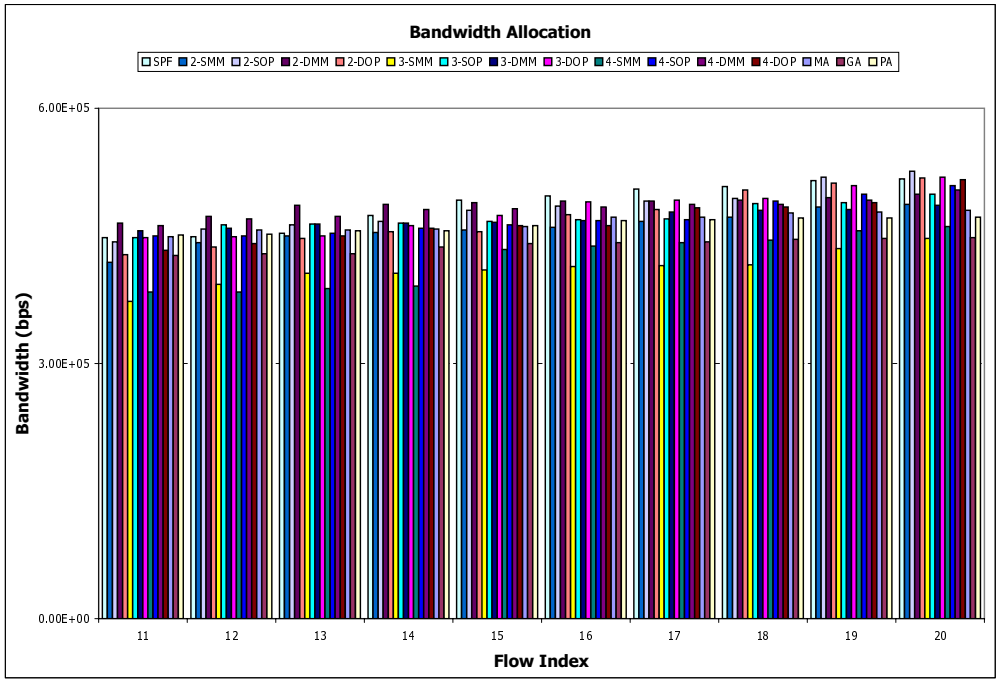

Figure 11: R50-Sparse - Bandwidth of flows 11–20.

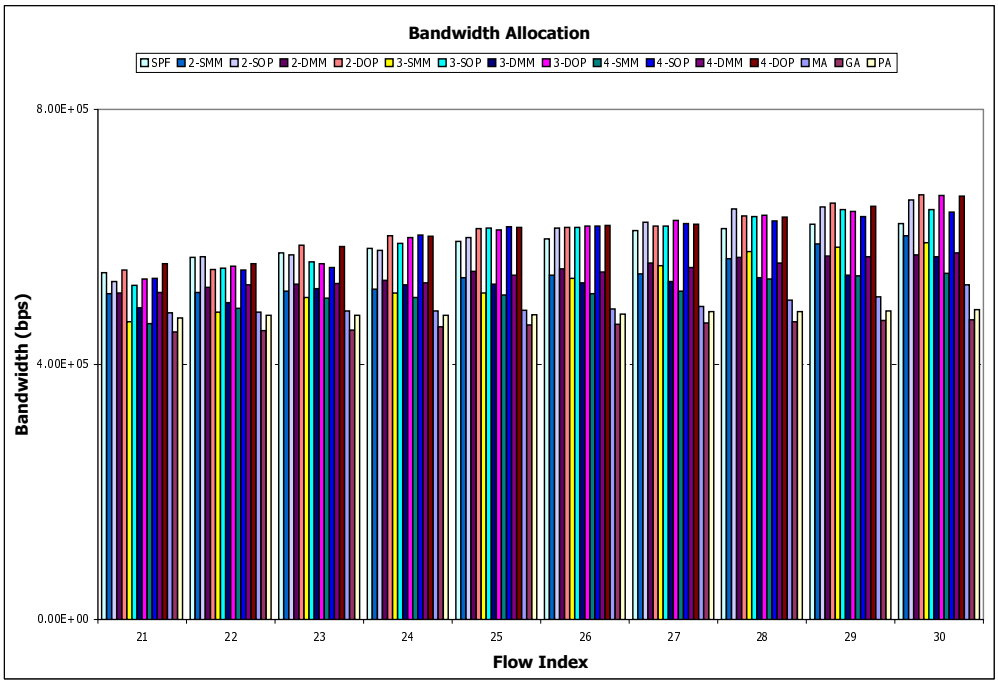

Figure 12: R50-Sparse - Bandwidth of flows 21–30.

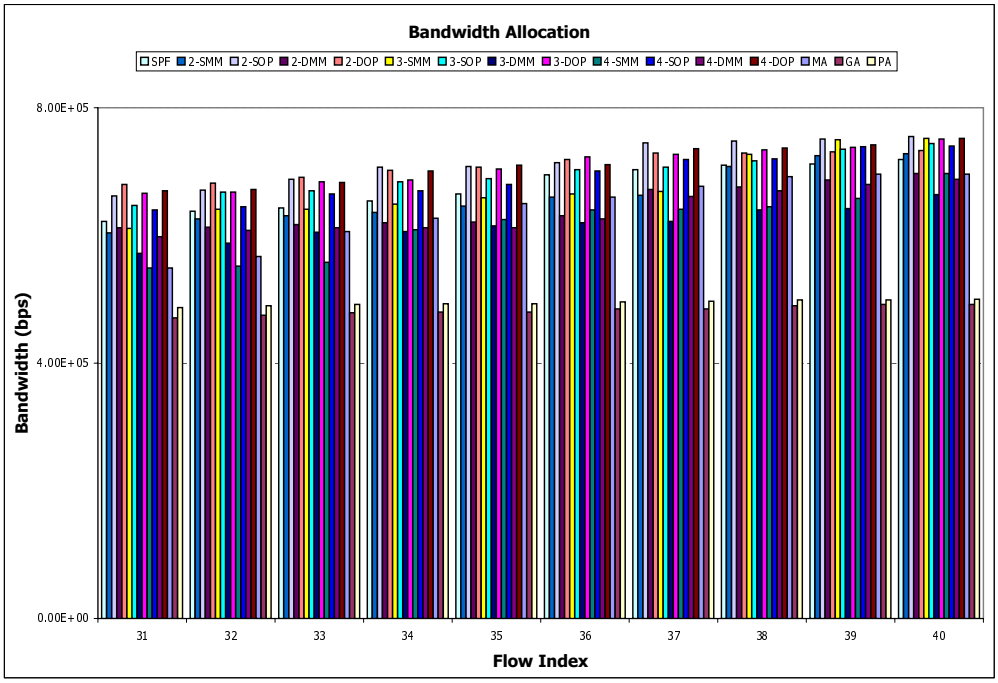

Figure 13: R50-Sparse - Bandwidth of flows 31–40.

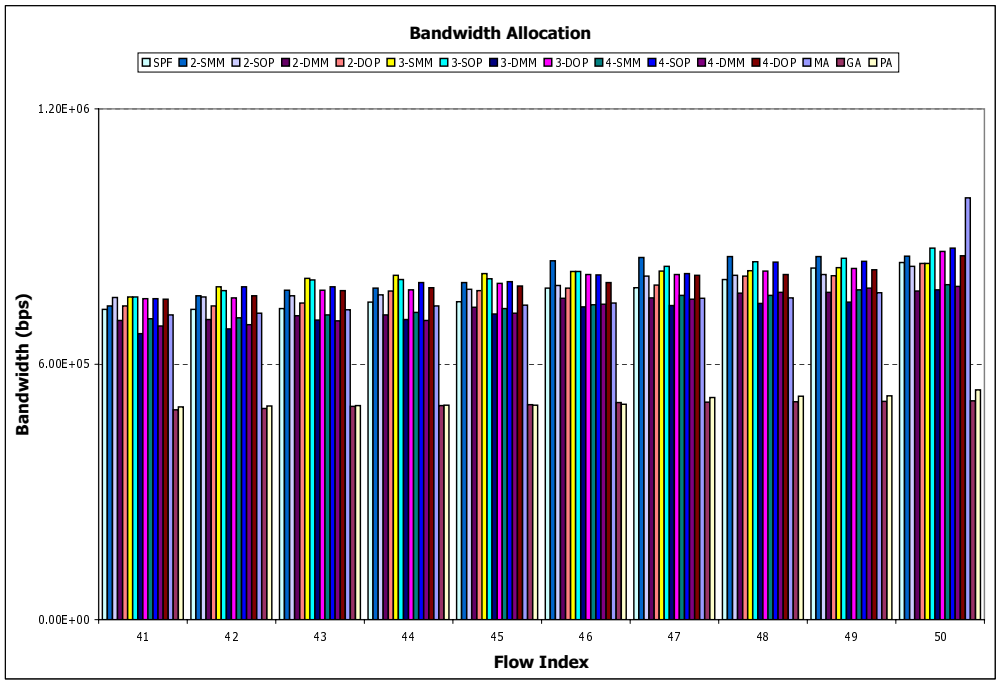

Figure 14: R50-Sparse - Bandwidth of flows 41–50.

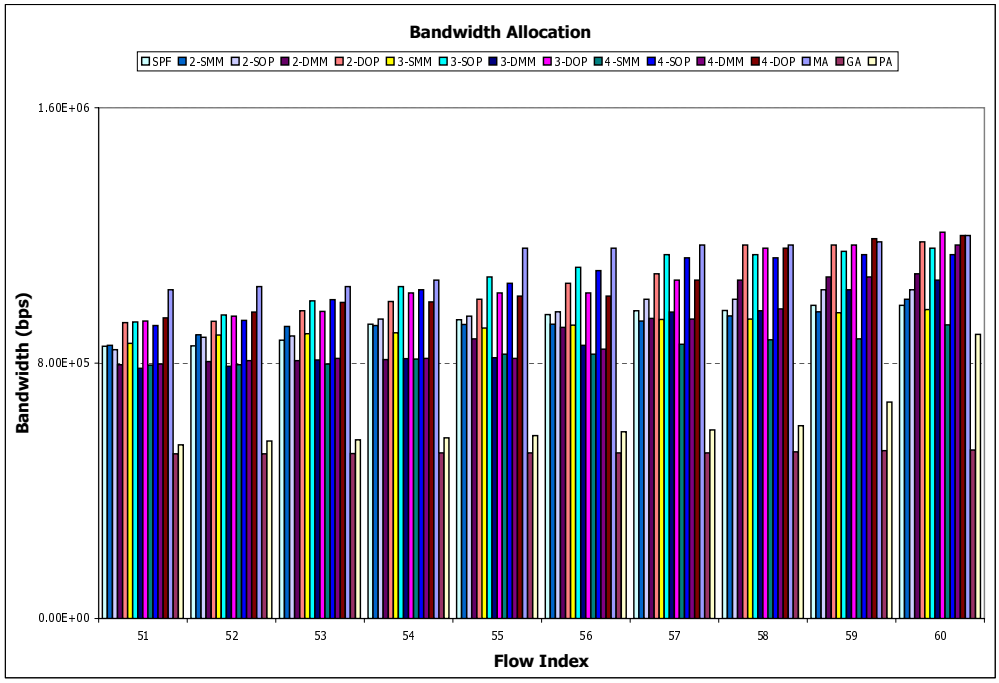

Figure 15: R50-Sparse - Bandwidth of flows 51–60.

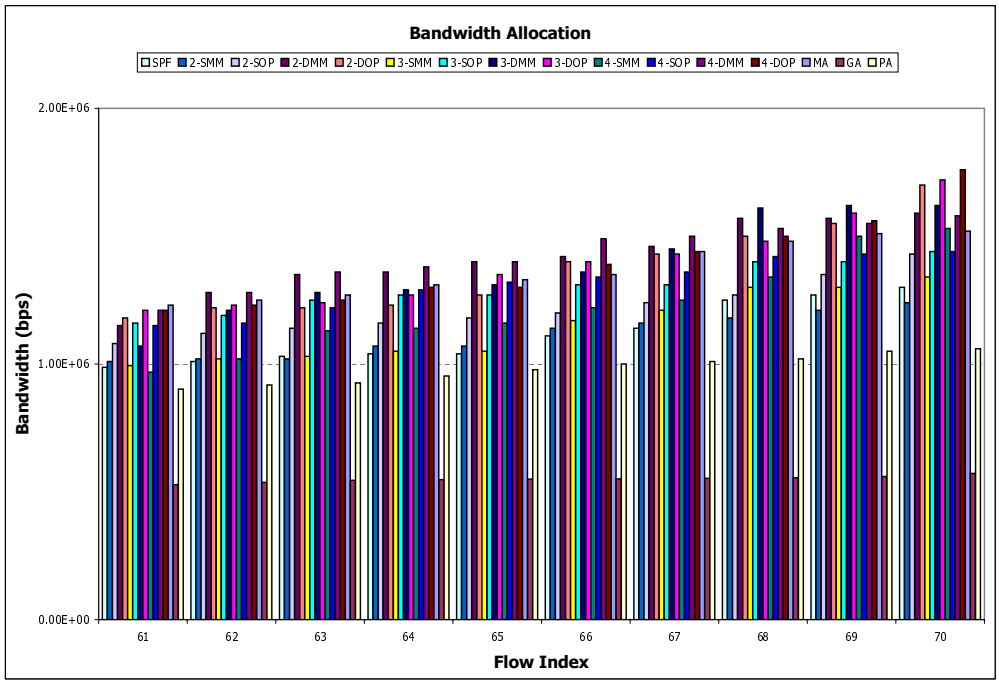

Figure 16: R50-Sparse - Bandwidth of flows 61–70.

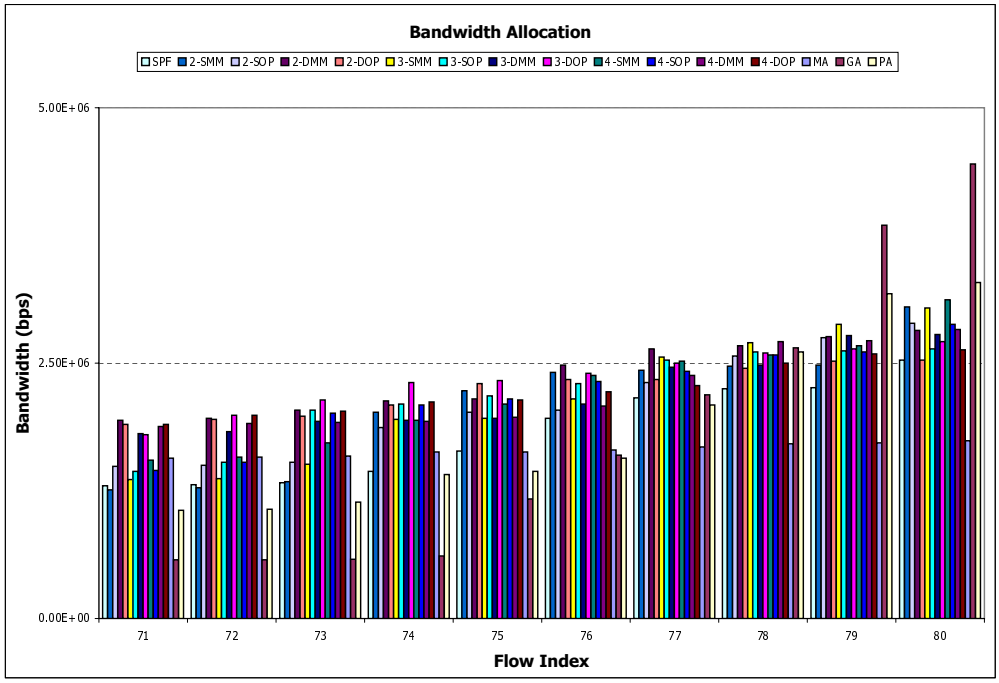

Figure 17: R50-Sparse - Bandwidth of flows 71–80.

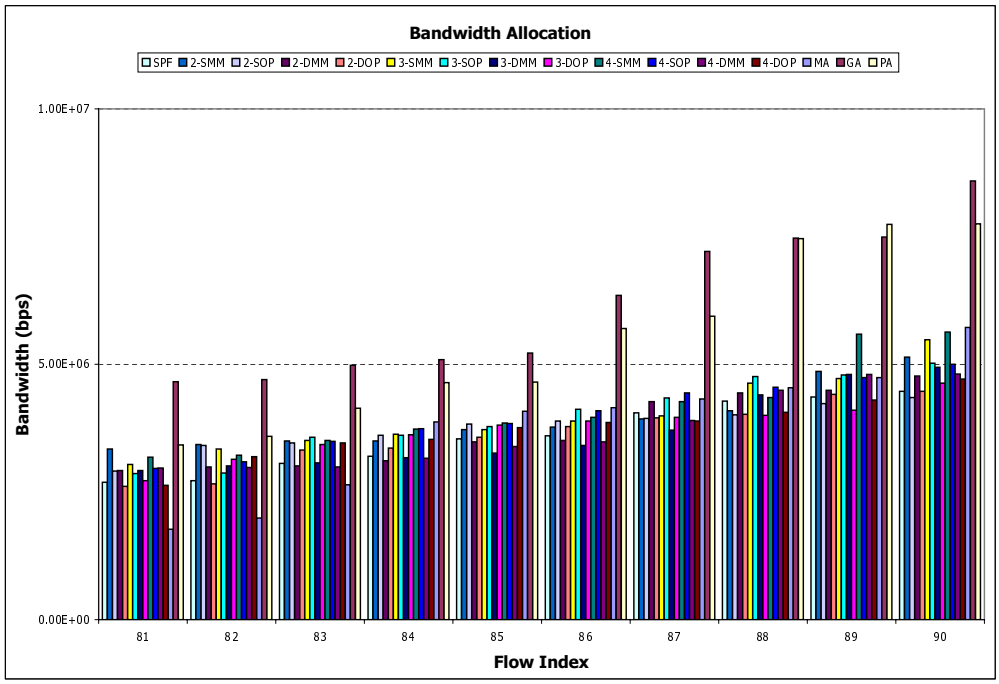

Figure 18: R50-Sparse - Bandwidth of flows 81–90.

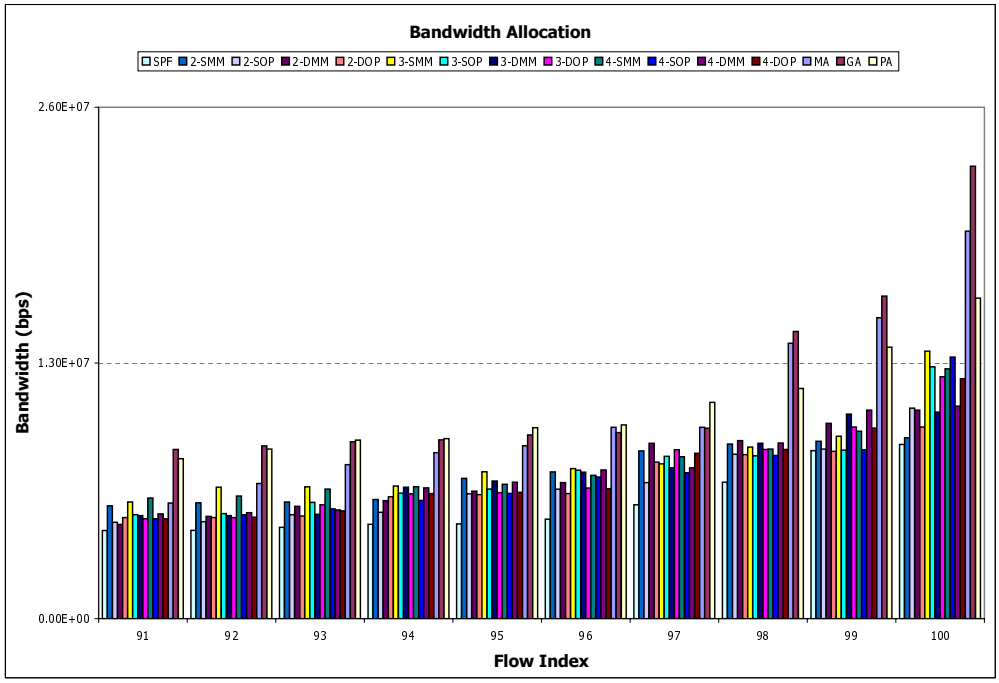

Figure 19: R50-Sparse - Bandwidth of flows 91–100.

50-Node Dense Random Topology

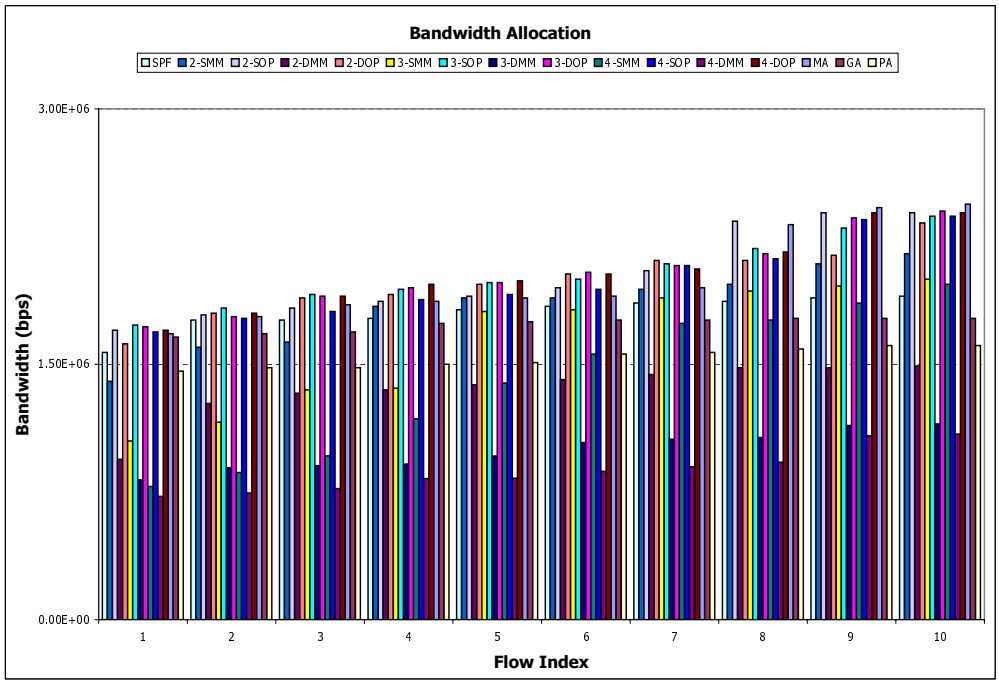

Figure 20: R50-Dense - Bandwidth of flows 1–10.

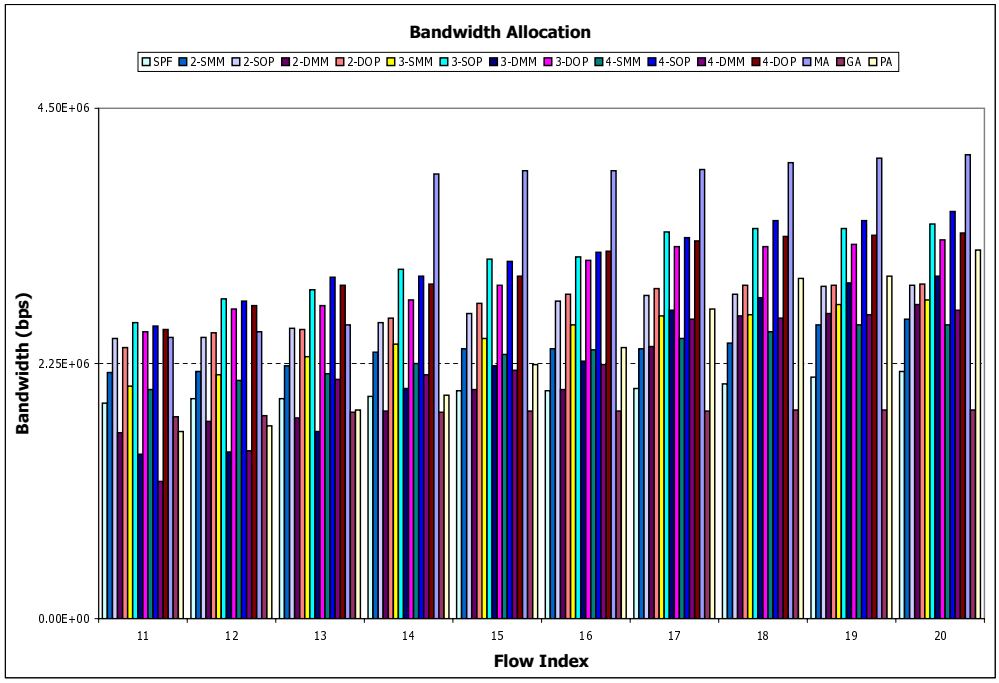

Figure 21: R50-Dense - Bandwidth of flows 11–20.

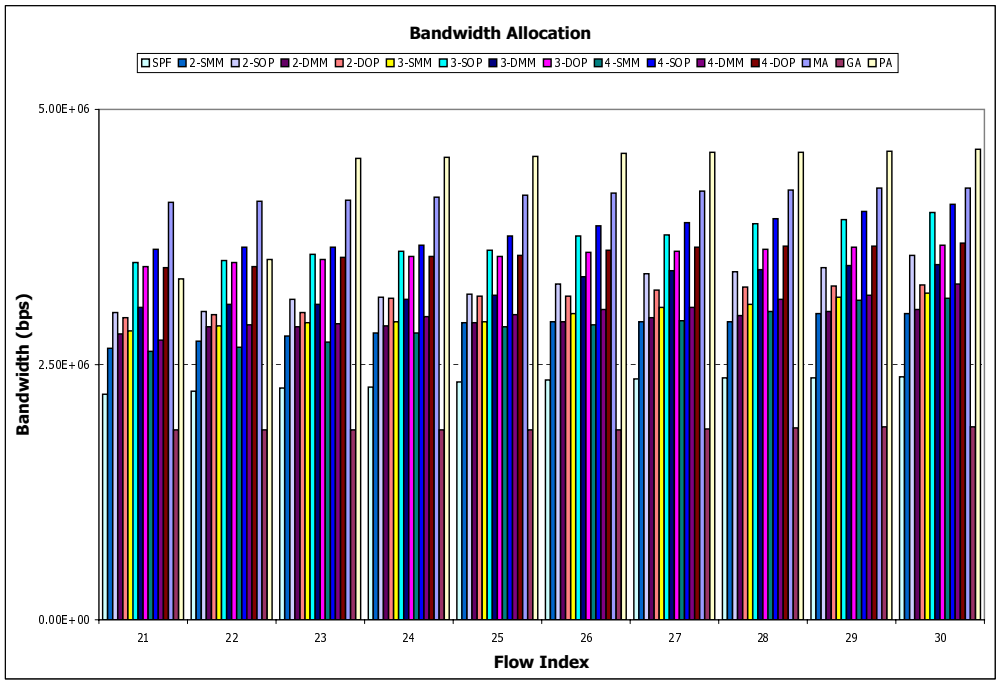

Figure 22: R50-Dense - Bandwidth of flows 21–30.

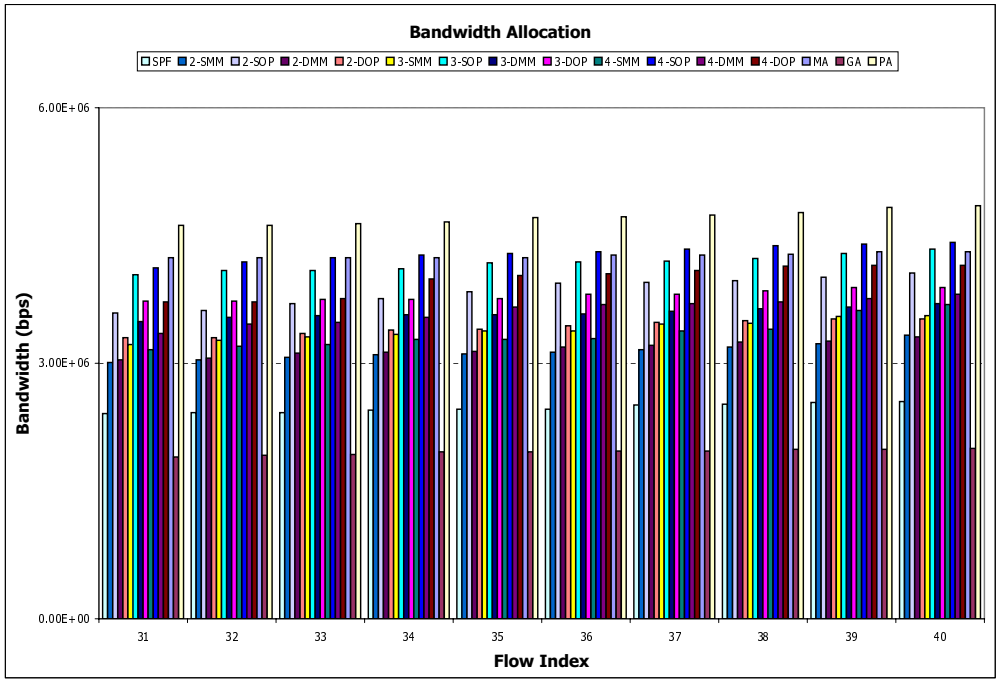

Figure 23: R50-Dense - Bandwidth of flows 31–40.

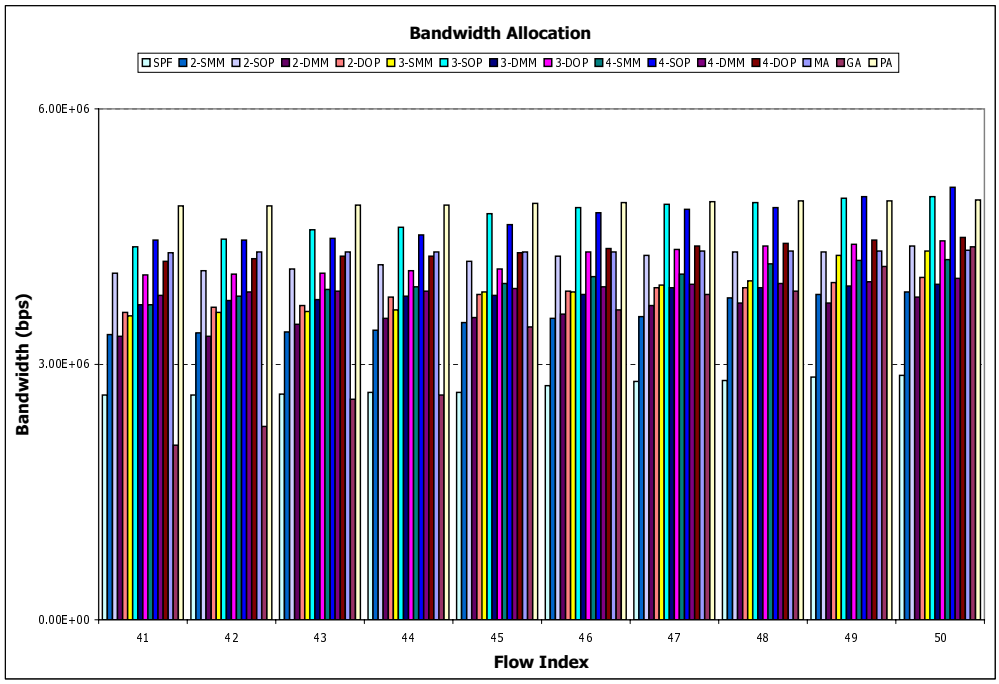

Figure 24: R50-Dense - Bandwidth of flows 41–50.

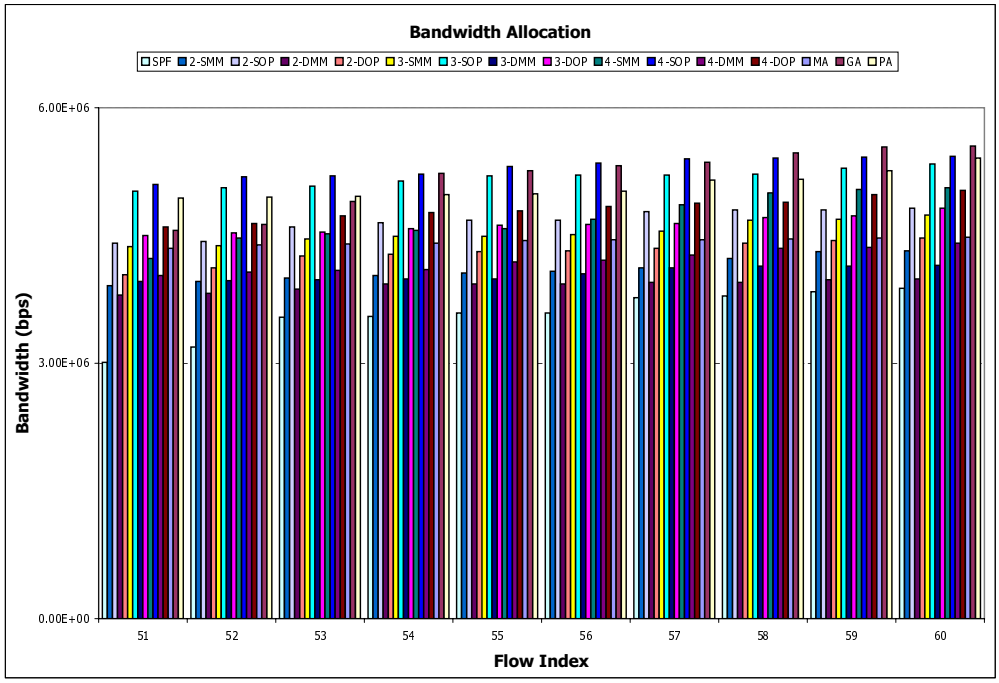

Figure 25: R50-Dense - Bandwidth of flows 51–60.

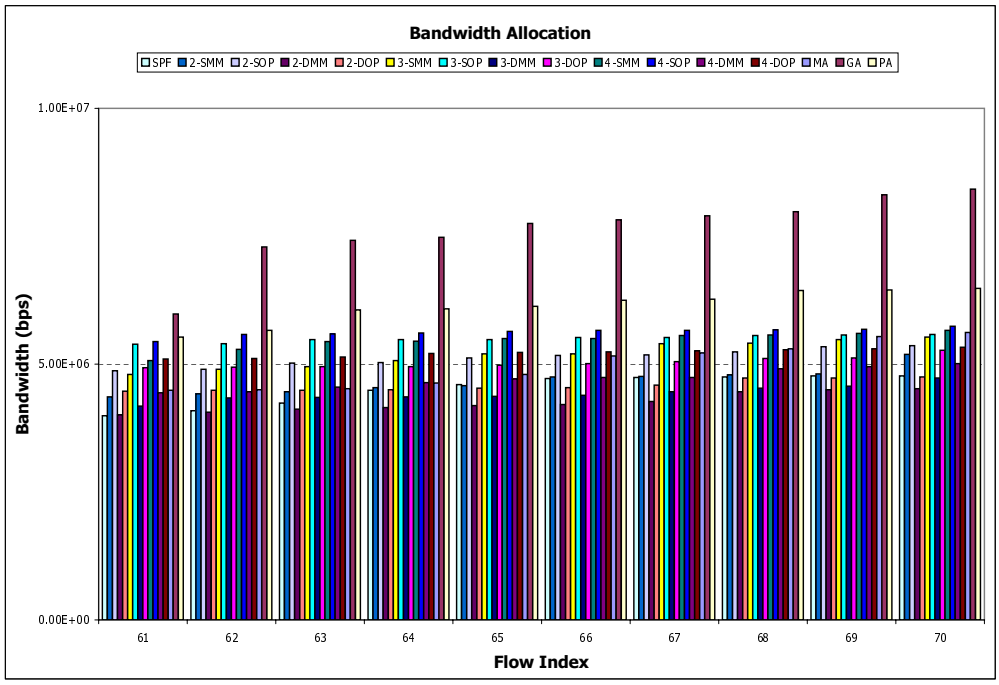

Figure 26: R50-Dense - Bandwidth of flows 61–70.

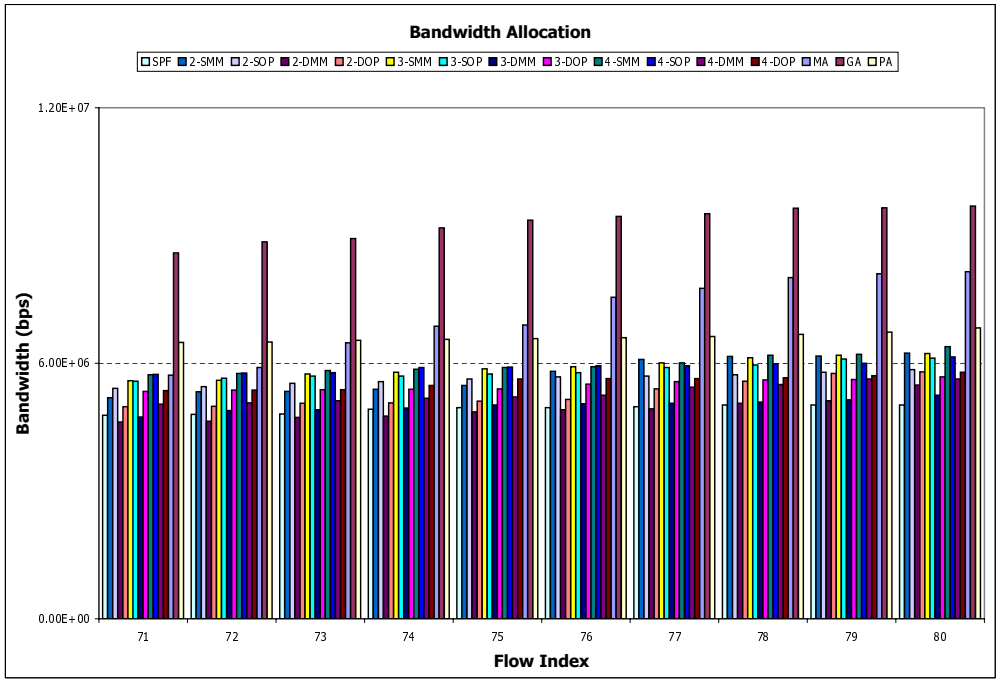

Figure 27: R50-Dense - Bandwidth of flows 71–80.

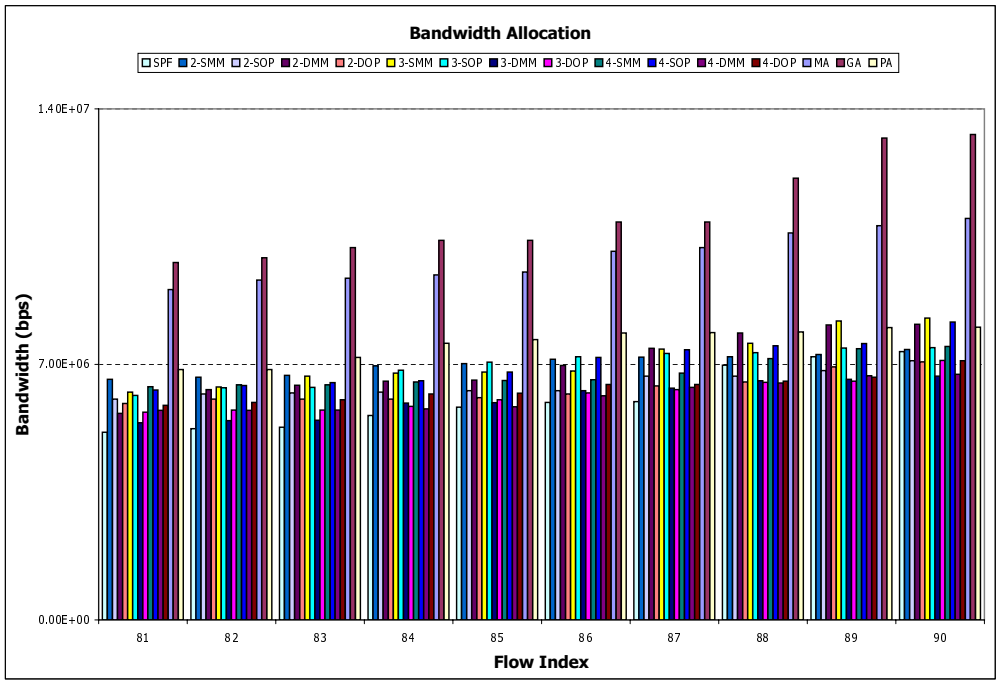

Figure 28: R50-Dense - Bandwidth of flows 81–90.

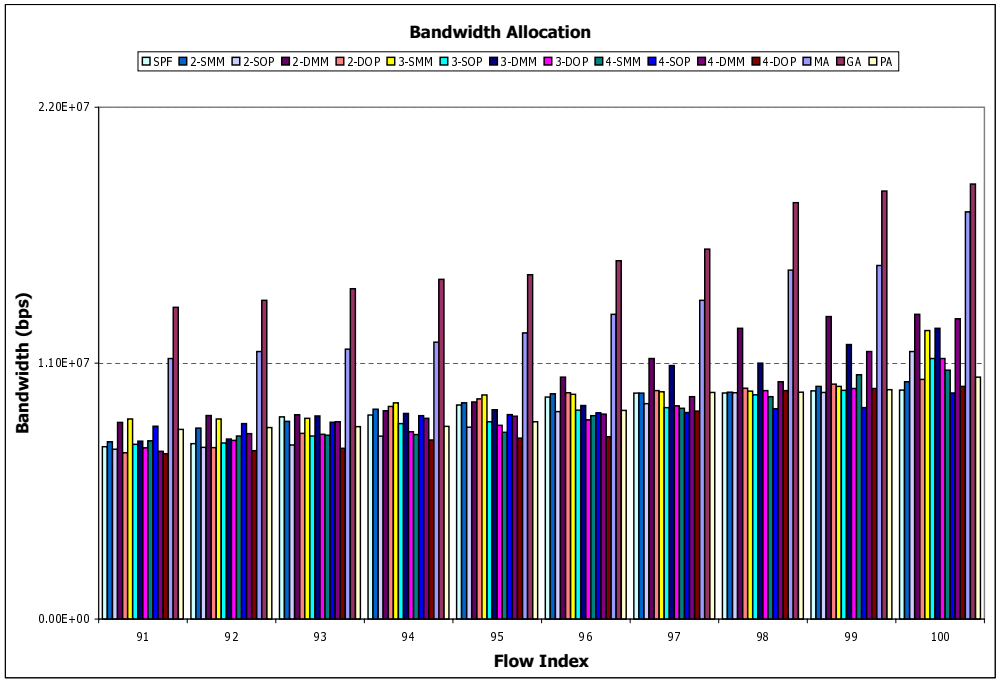

Figure 29: R50-Dense - Bandwidth of flows 91–100.

50-Node Sparse Transit-Stub Topology

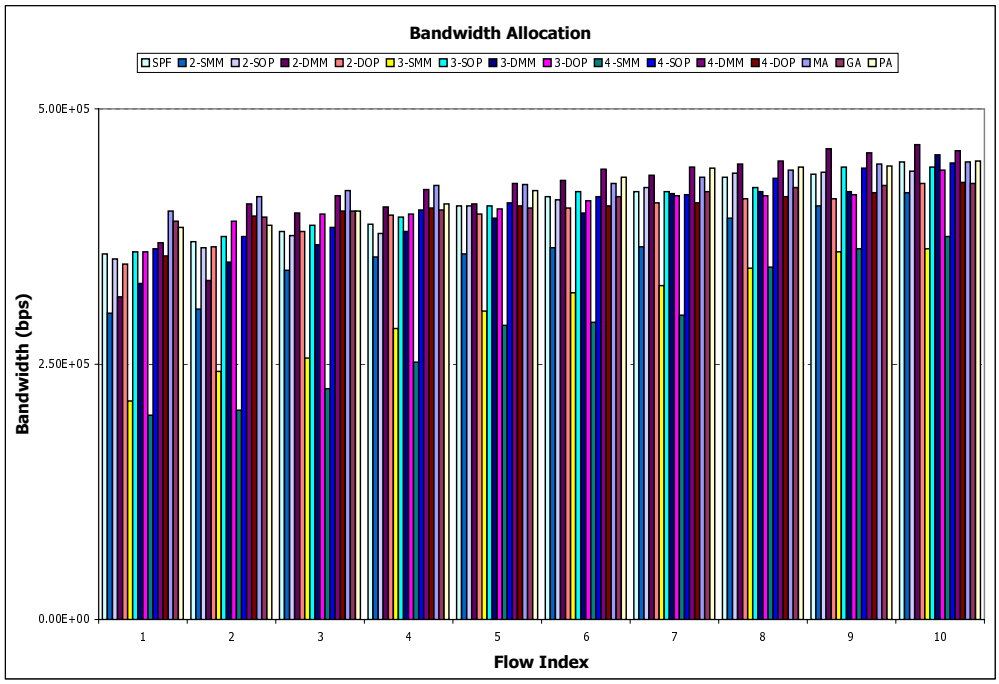

Figure 30: TS50-Sparse - Bandwidth of flows 1–10.

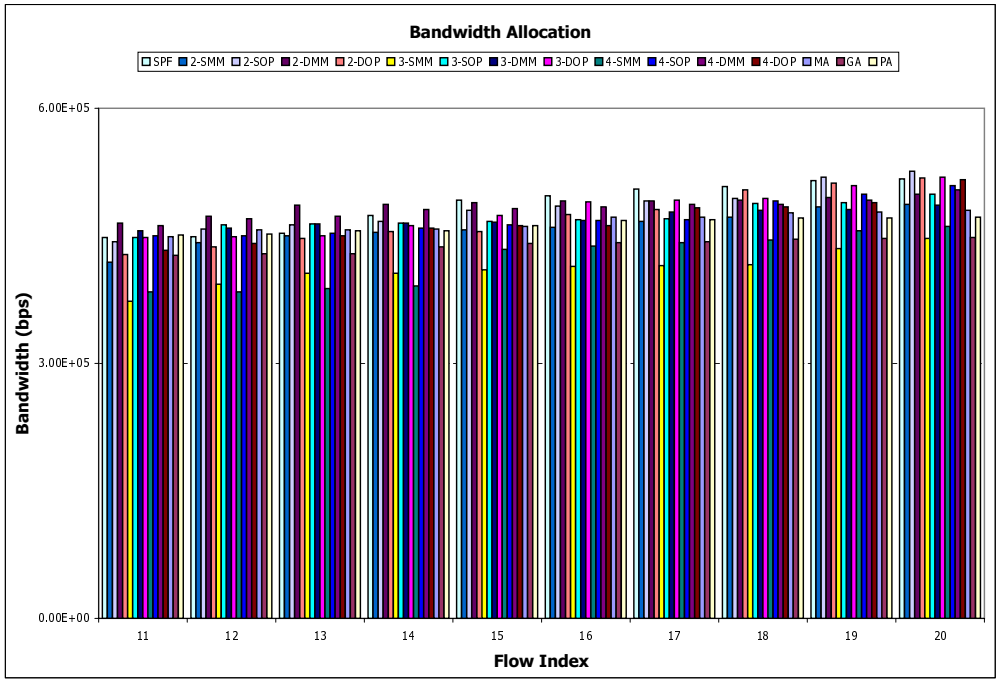

Figure 31: TS50-Sparse - Bandwidth of flows 11–20.

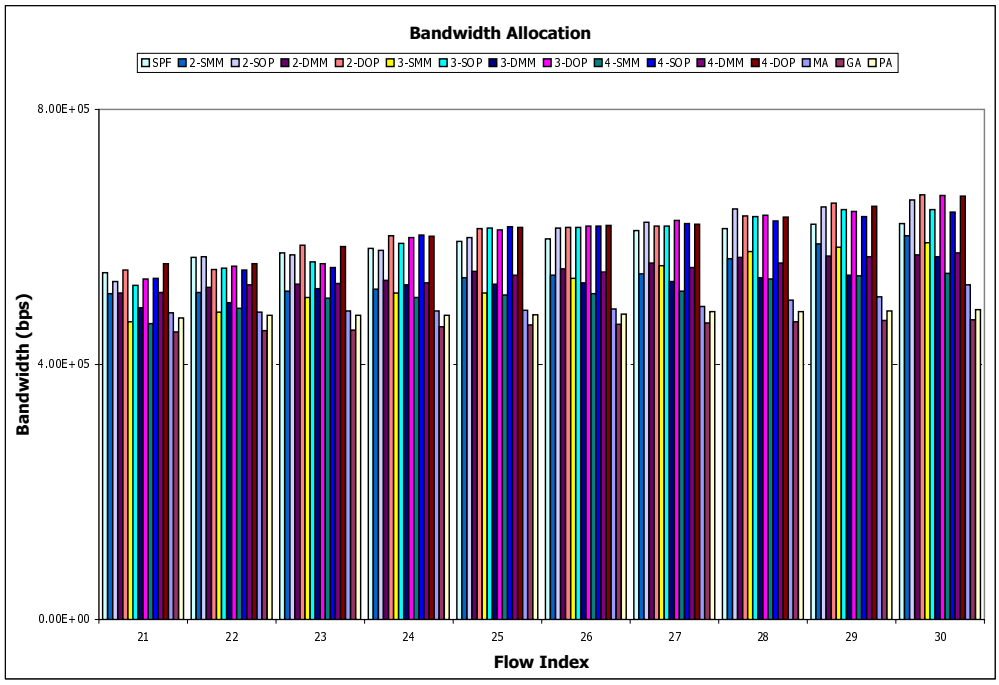

Figure 32: TS50-Sparse - Bandwidth of flows 21–30.

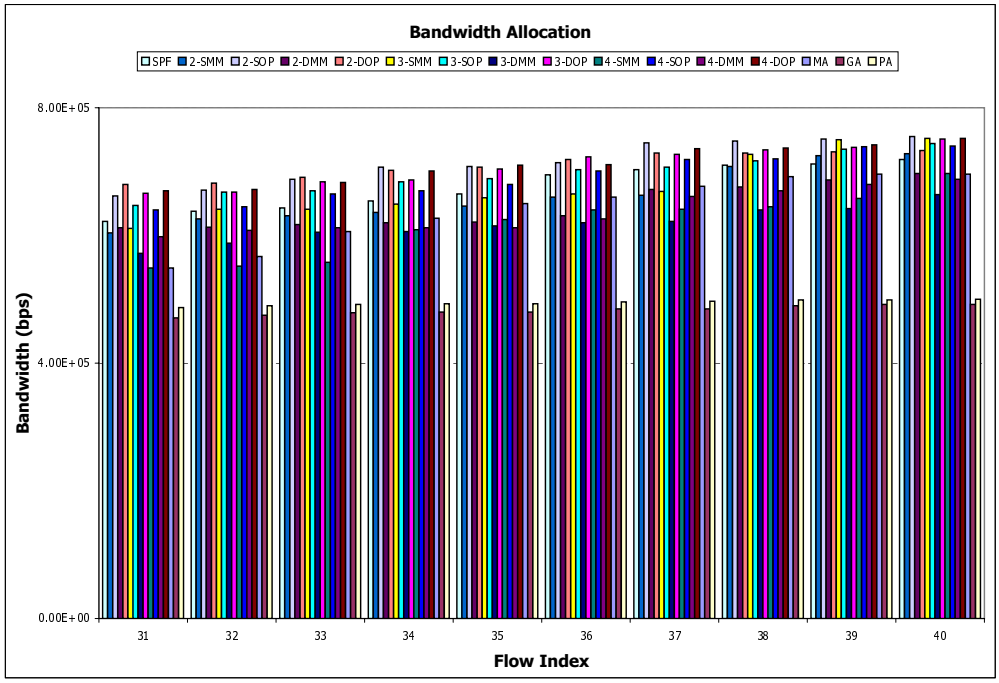

Figure 33: TS50-Sparse - Bandwidth of flows 31–40.

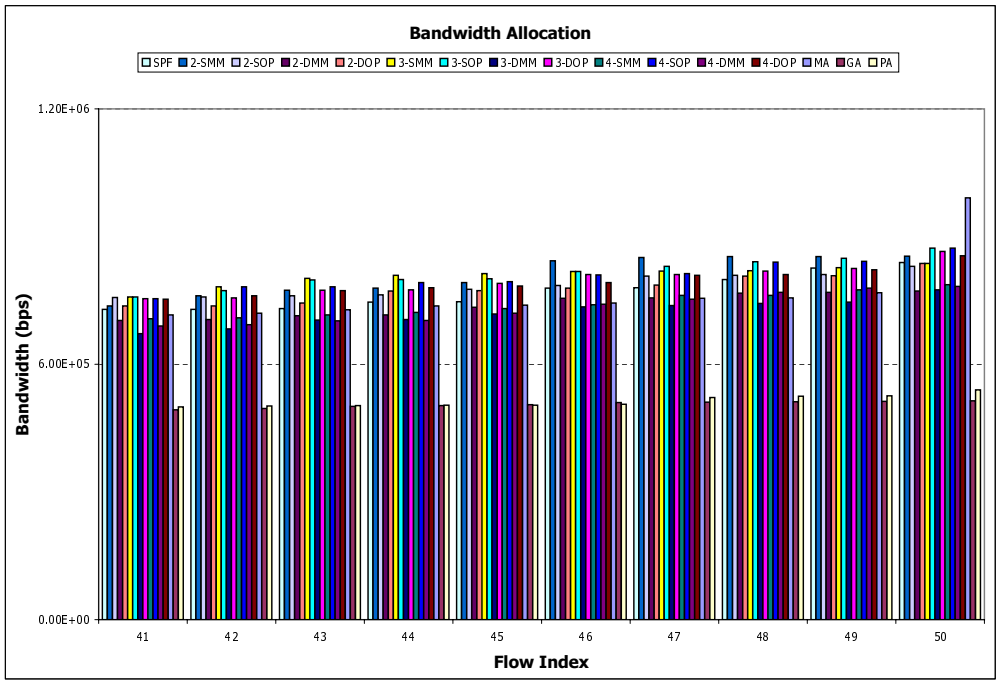

Figure 34: TS50-Sparse - Bandwidth of flows 41–50.

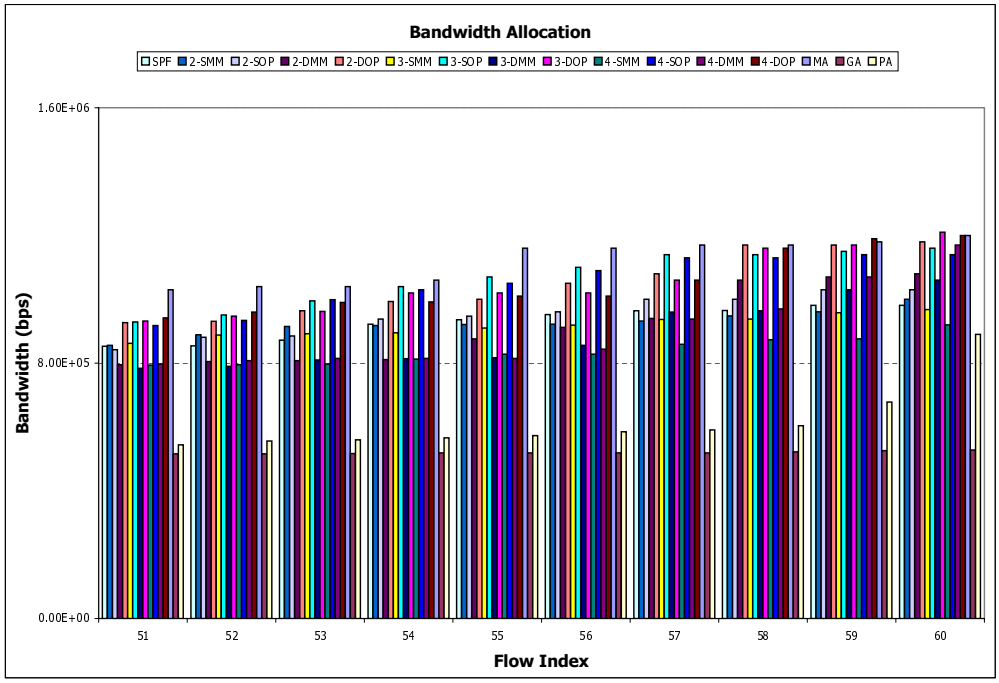

Figure 35: TS50-Sparse - Bandwidth of flows 51–60.

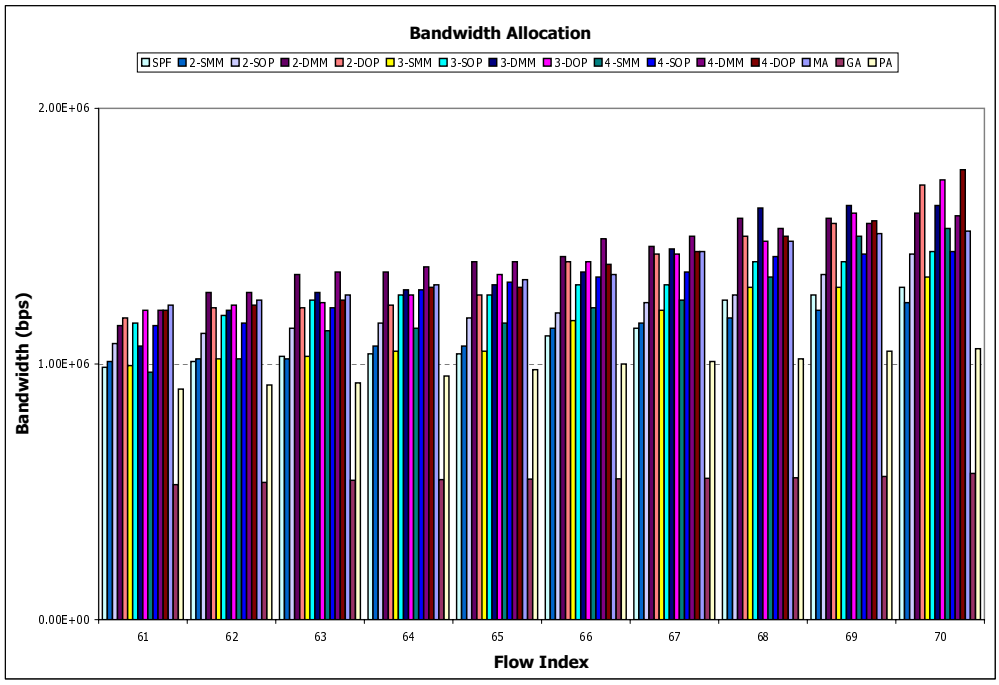

Figure 36: TS50-Sparse - Bandwidth of flows 61–70.

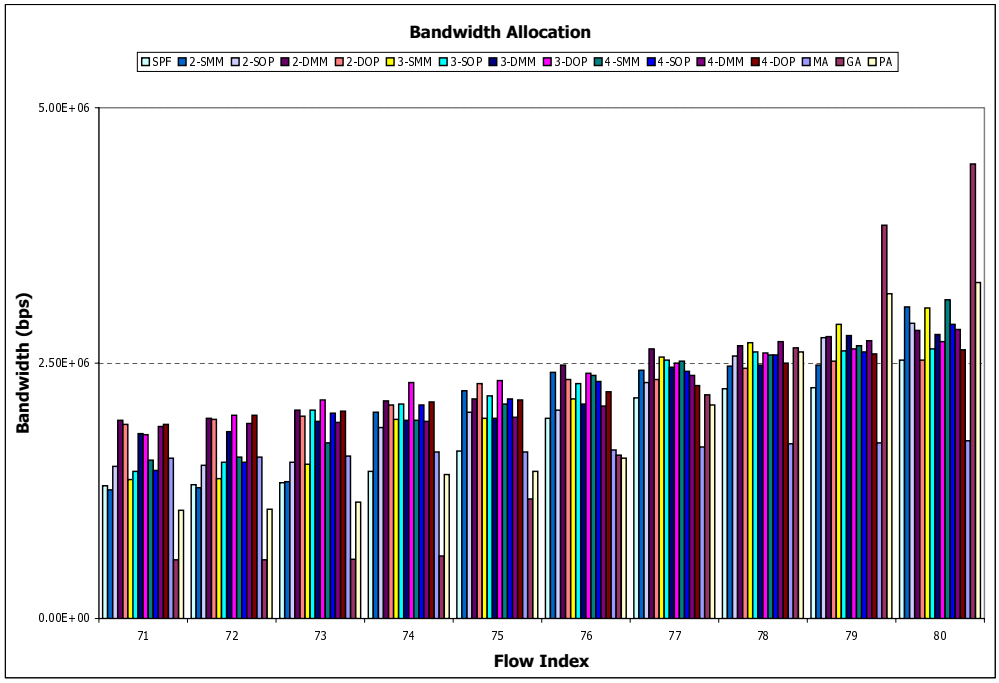

Figure 37: TS50-Sparse - Bandwidth of flows 71–80.

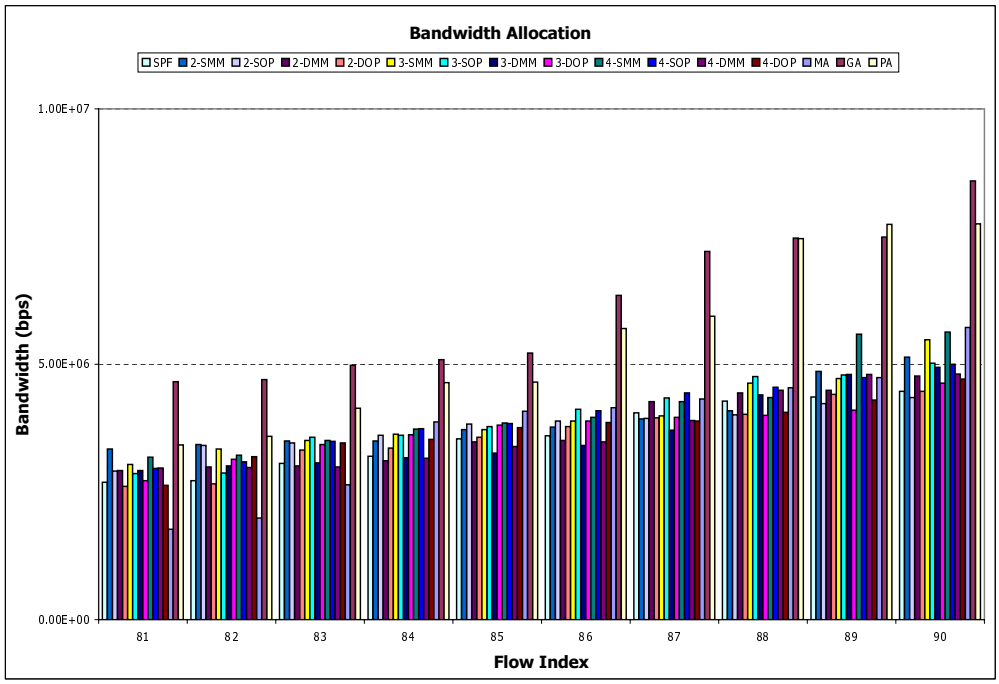

Figure 38: TS50-Sparse - Bandwidth of flows 81–90.

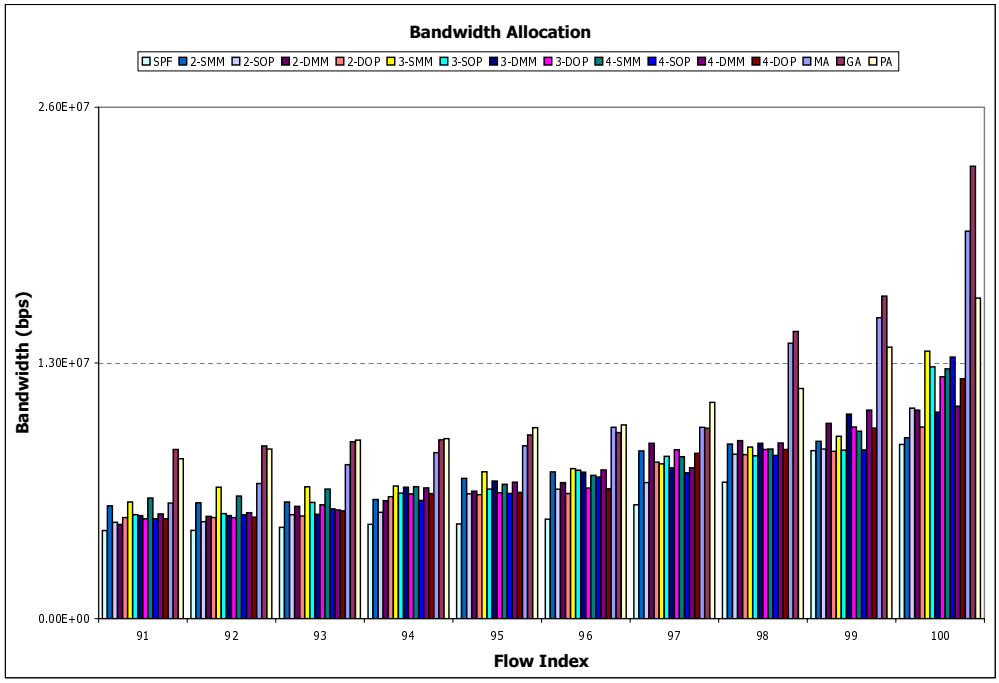

Figure 39: TS50-Sparse - Bandwidth of flows 91–100.

50-Node Dense Transit-Stub Topology

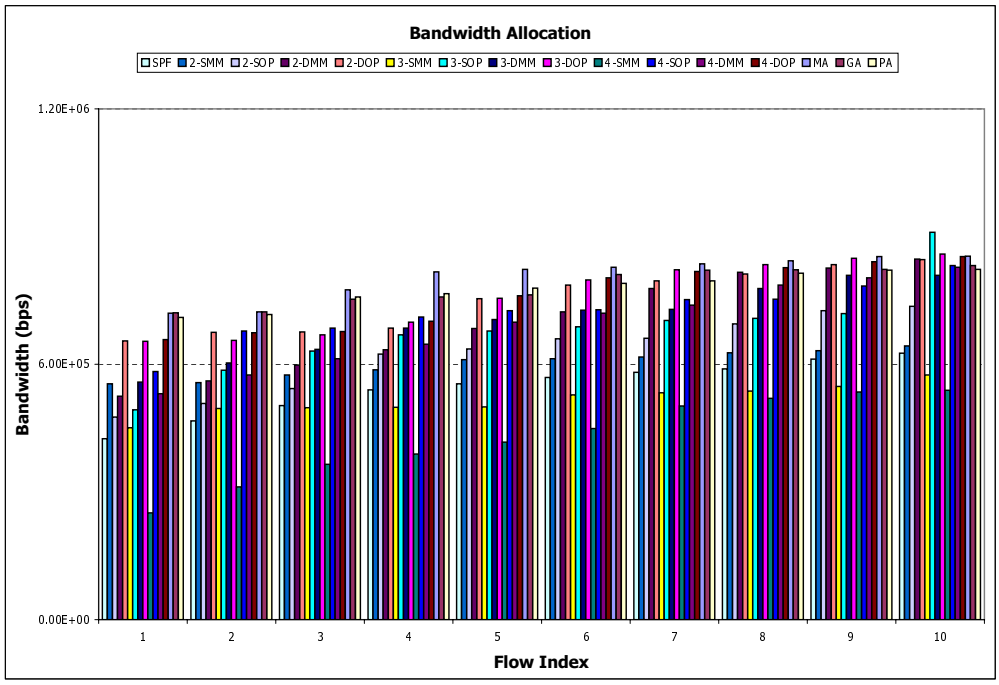

Figure 40: TS50-Dense - Bandwidth of flows 1–10.

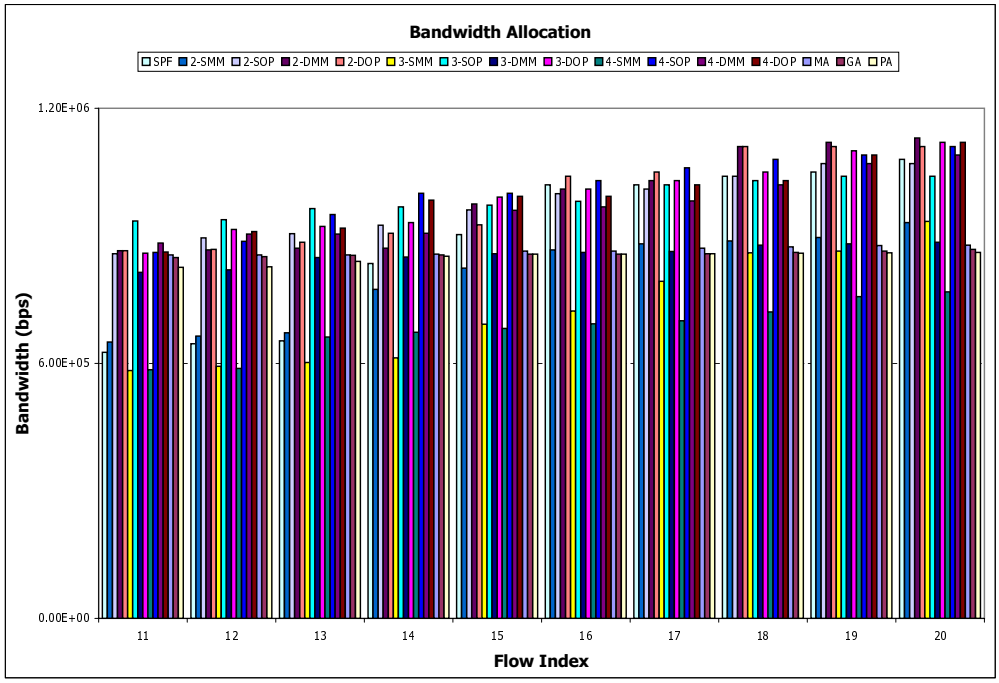

Figure 41: TS50-Dense - Bandwidth of flows 11–20.

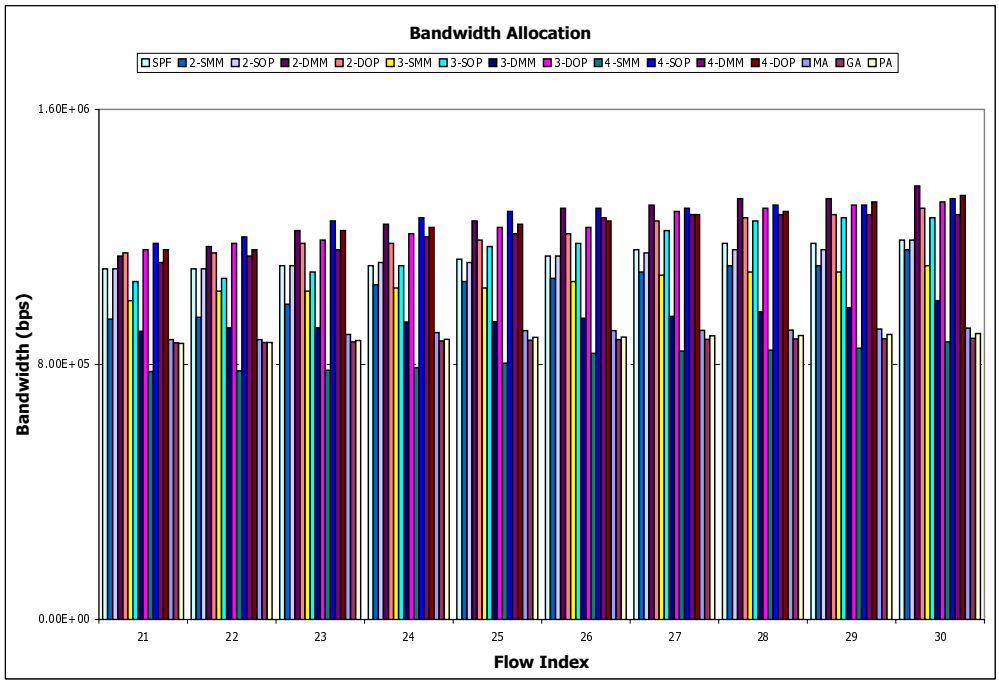

Figure 42: TS50-Dense - Bandwidth of flows 21–30.

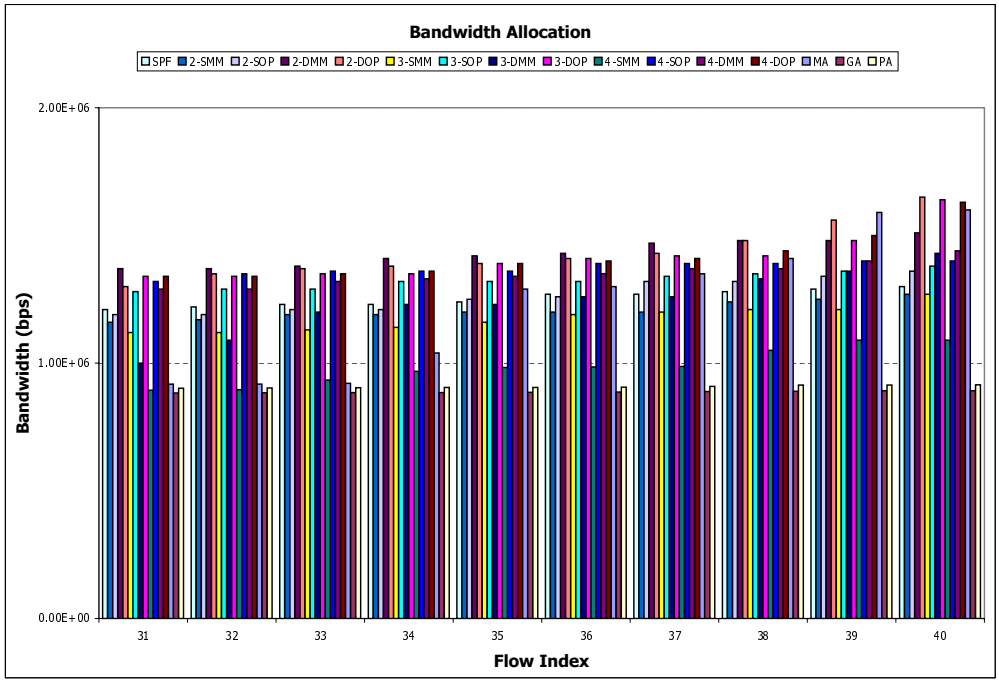

Figure 43: TS50-Dense - Bandwidth of flows 31–40.

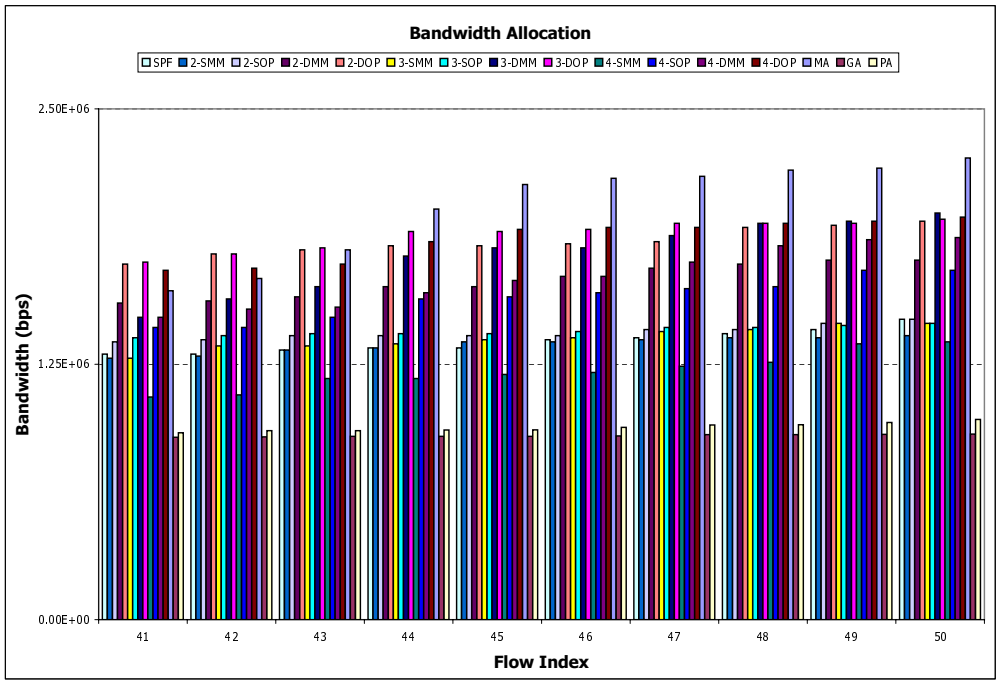

Figure 44: TS50-Dense - Bandwidth of flows 41–50.

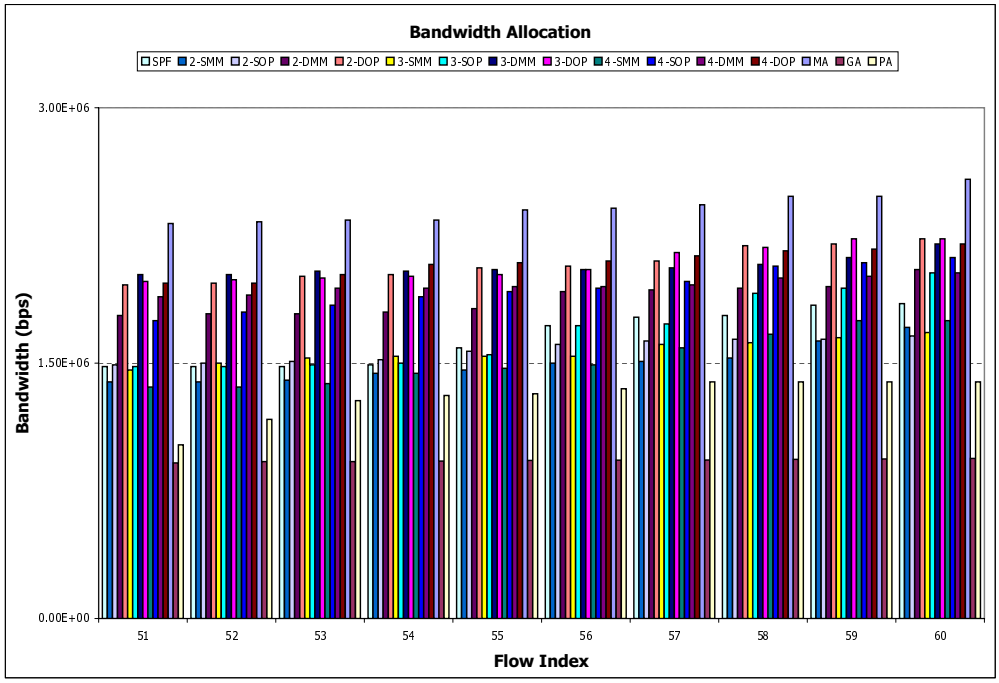

Figure 45: TS50-Dense - Bandwidth of flows 51–60.

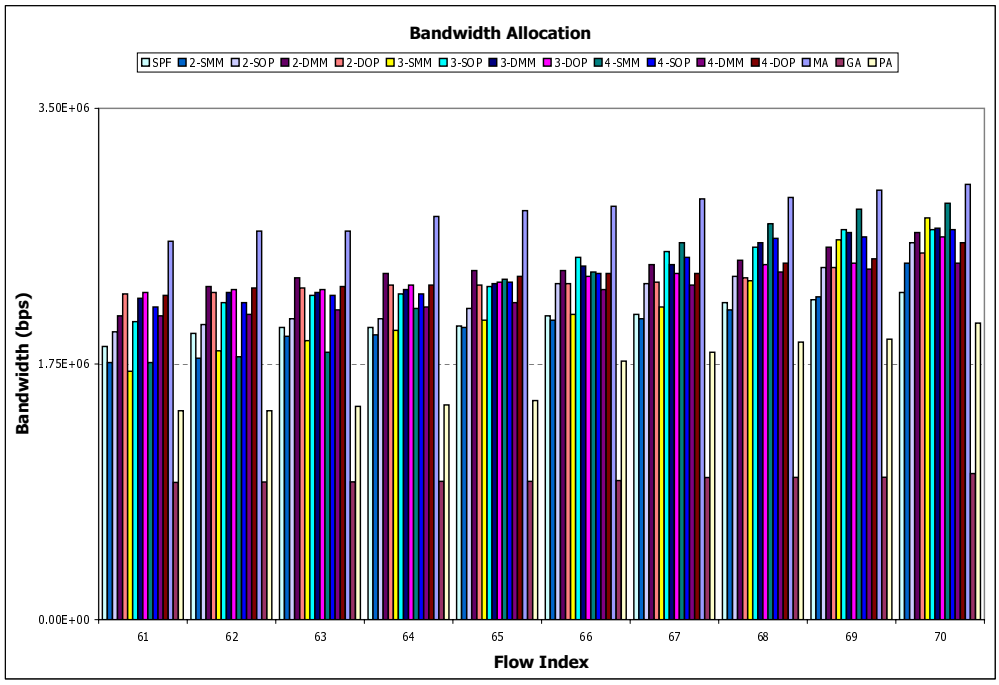

Figure 46: TS50-Dense - Bandwidth of flows 61–70.

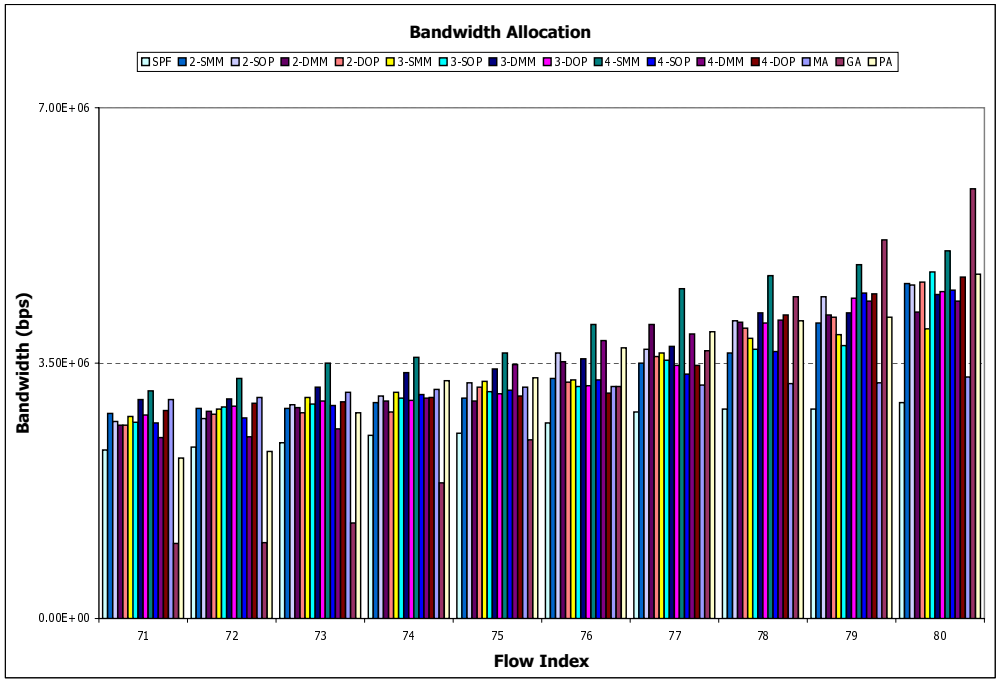

Figure 47: TS50-Dense - Bandwidth of flows 71–80.

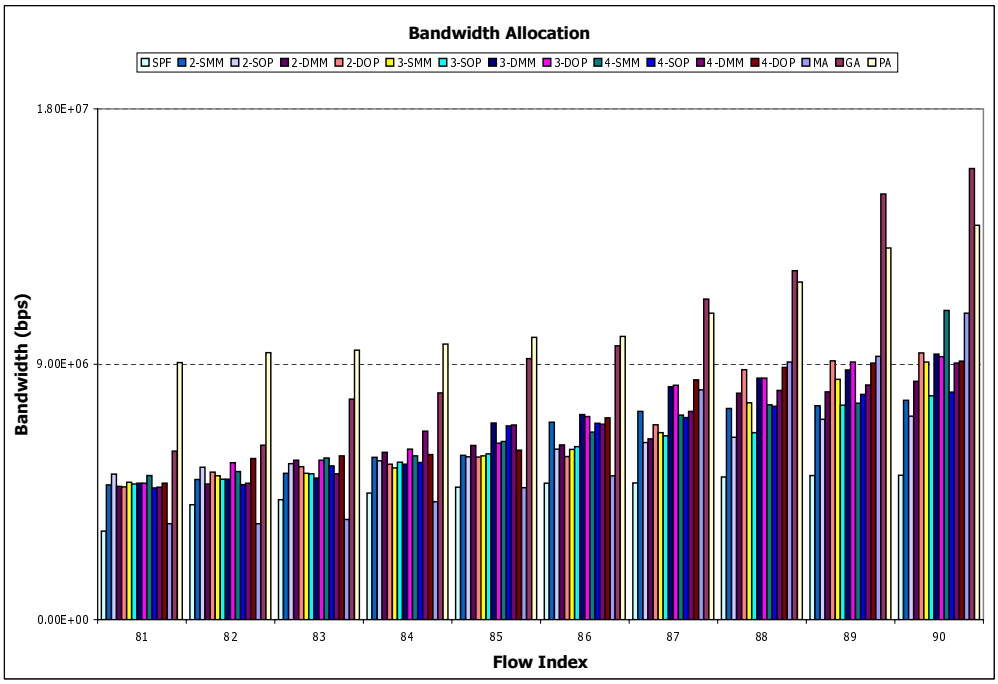

Figure 48: TS50-Dense - Bandwidth of flows 81–90.

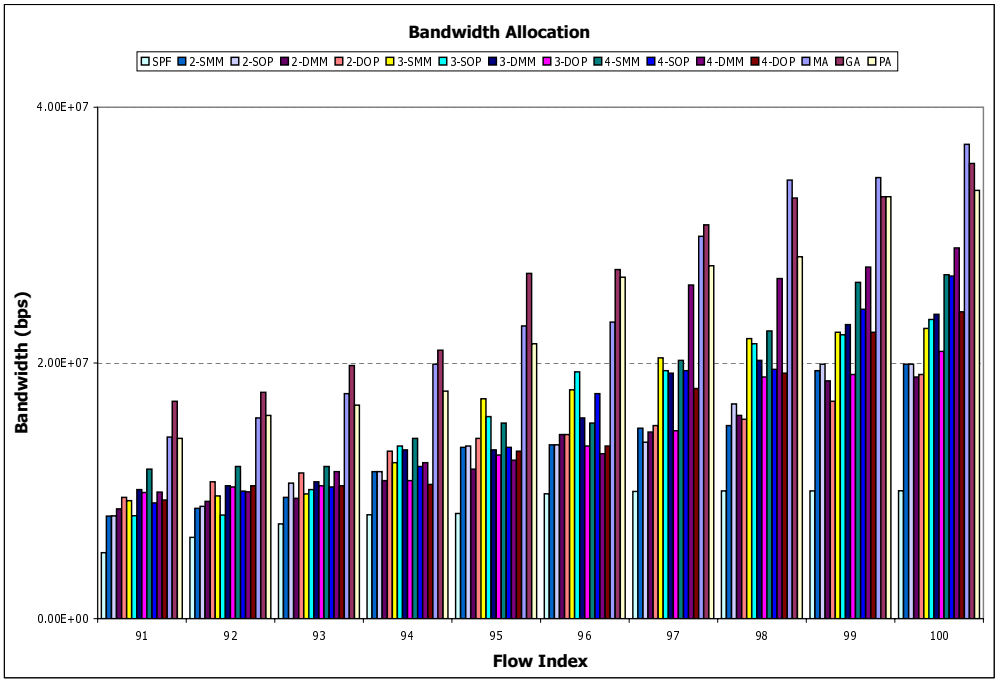

Figure 49: TS50-Dense - Bandwidth of flows 91–100.

Sprint Backbone Topology

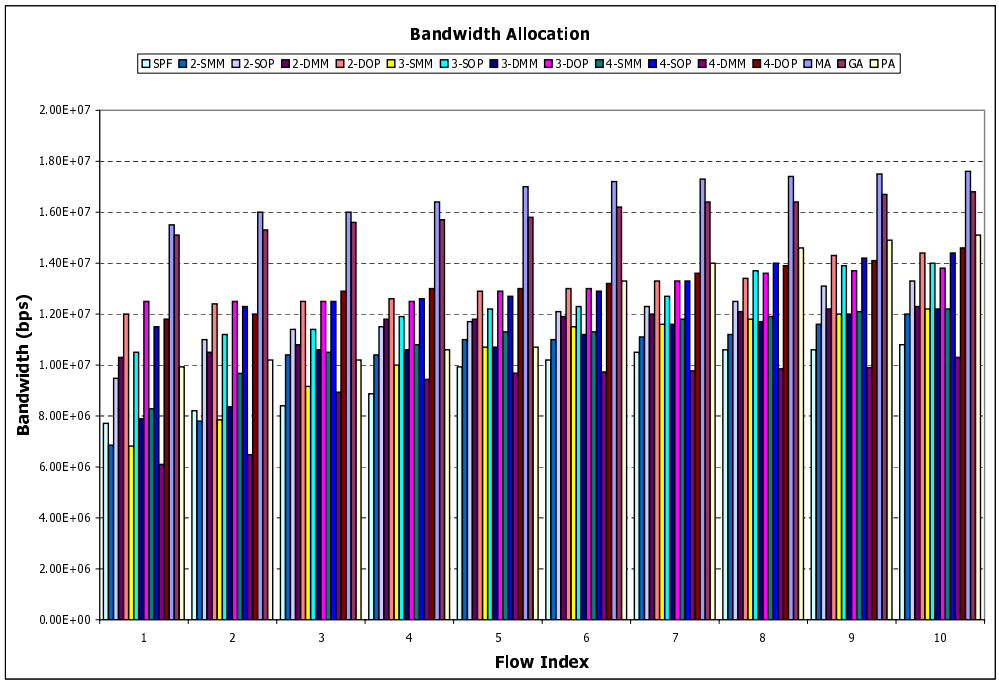

Figure 50: Sprint Backbone - Bandwidth of flows 1–10.

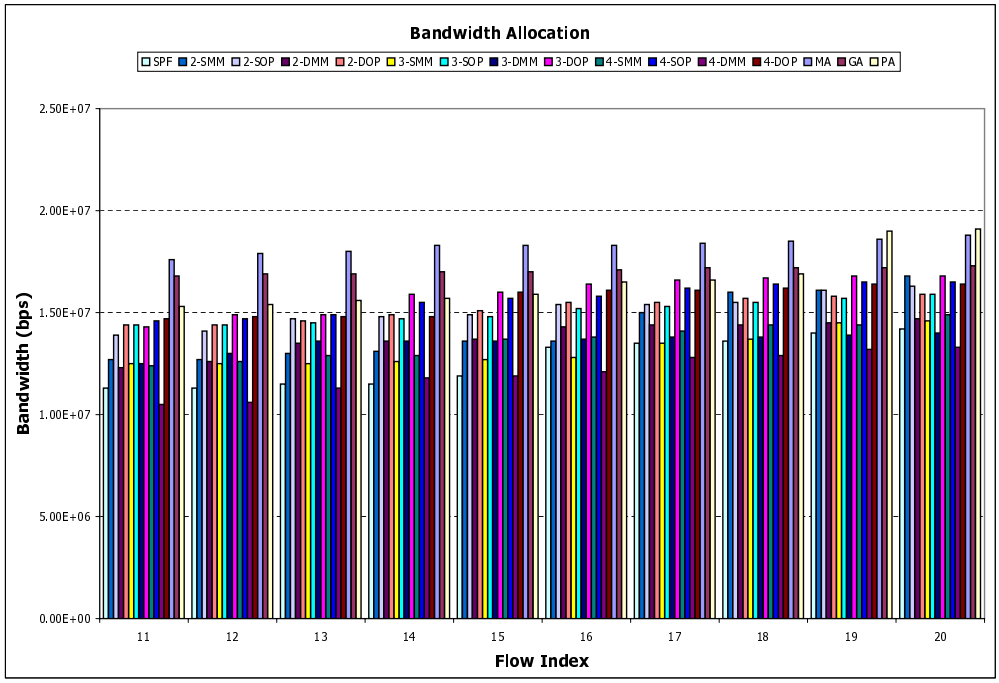

Figure 51: Sprint Backbone - Bandwidth of flows 11–20.

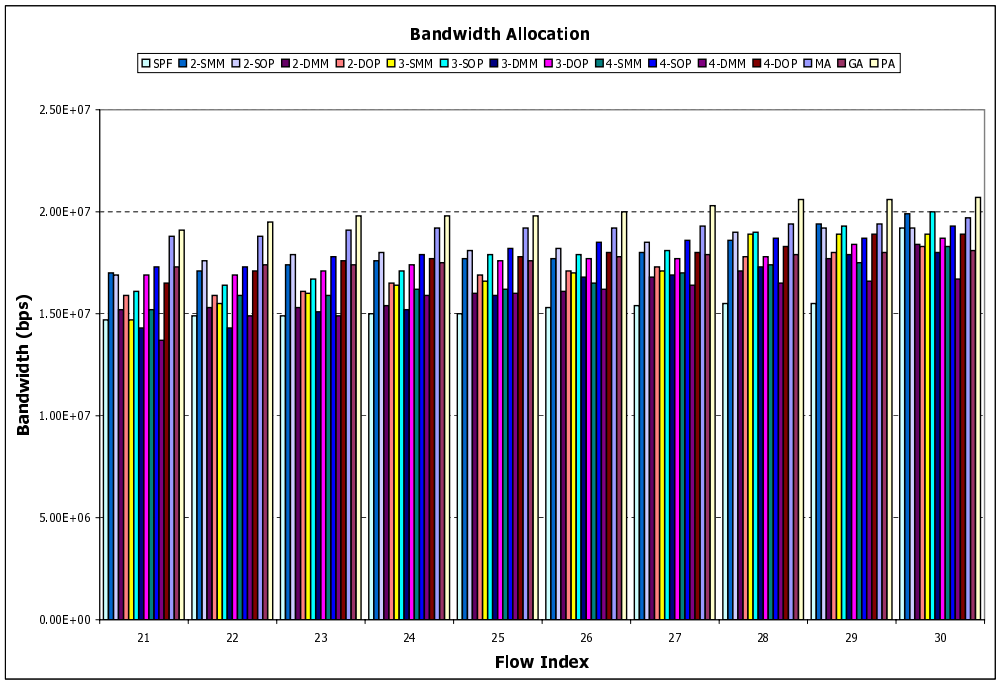

Figure 52: Sprint Backbone - Bandwidth of flows 21–30.

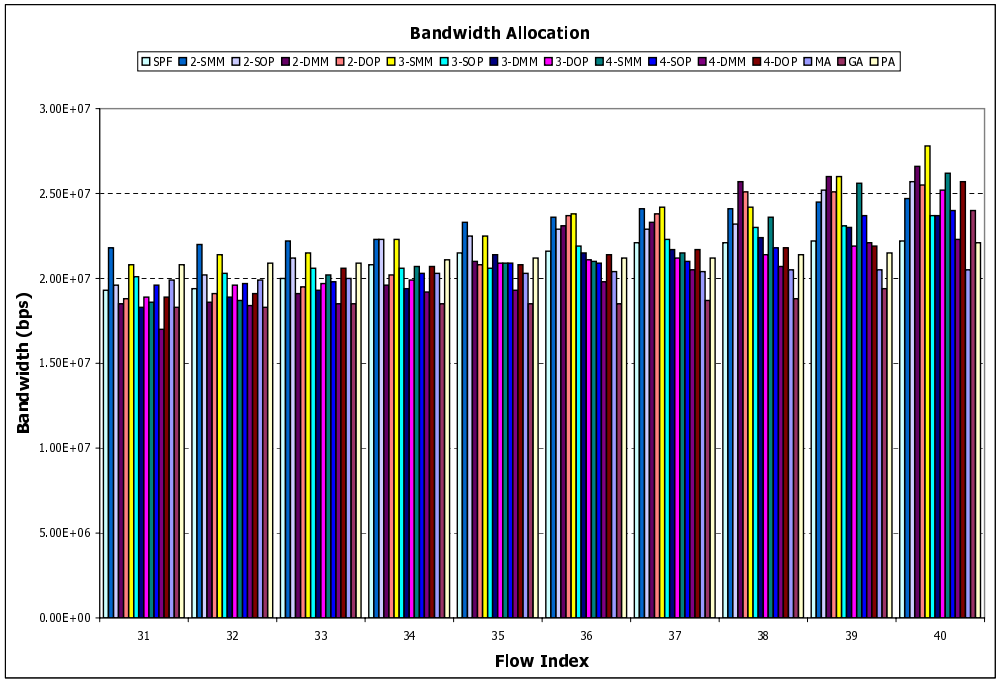

Figure 53: Sprint Backbone - Bandwidth of flows 31–40.

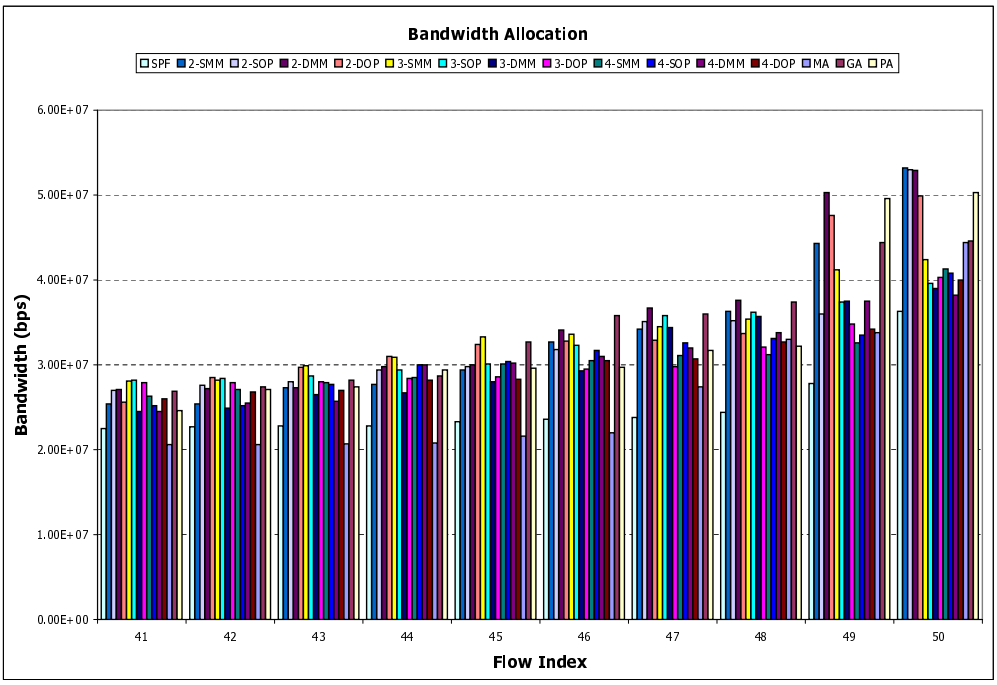

Figure 54: Sprint Backbone - Bandwidth of flows 41–50.

REFERENCES

[1] "LAN emulation over ATM - version 1." ATM Forum, Jan. 1995.

[2] "Multiprotocol over ATM - version 1." ATM Forum, July 1997.

[3] Athuraliya, S., Li, V. H., Low, S. H., and Yin, Q., "REM: Active queue management," *IEEE Network*, pp. 48–53, May 2001.

[4] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and Swallow, G., "RSVP-TE: Extensions to RSVP for LSP tunnels." IETF RFC 3209, Dec. 2001.

[5] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and McManus, J., "Requirements for traffic engineering over MPLS." IETF RFC 3209, Jan. 2002.

[6] Bertsekas, D. and Gallager, R., *Data Networks*. Prentice Hall, 1992.

[7] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W., "An architecture for differentiated services." IETF RFC 2475, Dec. 1998.

[8] Braden, R., Clark, D., and Shenker, S., "Integrated services in the internet architecture: an overview." IETF RFC 1633, June 1994.

[9] Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S., "Resource reservation protocol (RSVP) - version 1 functional specification." IETF RFC 2205, Sept. 1997.

[10] Coffman, K. G. and Odlyzko, A. M., "Internet growth: Is there a "Moore's law" for data traffic?," in *Handbook of Massive Data Sets* (Abello, J., Pardalos, P. M., and Resende, M. G. C., eds.), pp. 47–93, Kluwer, 2002.

[11] Davie, B., Charny, A., Bennett, J., Benson, K., Boudec, J. L., Courtney, W., Davari, S., Firoiu, V., and Stiliadis, D., "An expedited forwarding PHB (per-hop behavior)." IETF RFC 3246, Mar. 2002.

[12] Floyd, S. and Jacobson, V., "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–?413, Aug. 1993.

[13] Fortz, B. and Thorup, M., "Internet traffic engineering by optimizing OSPF weights," in *Proceedings of INFOCOM 2000*, pp. 519–528, Mar. 2000.

[14] Girish, M., Zhou, B., , and Hu, J., "Formulation of the traffic engineering problems in MPLS based IP networks," in *Proceedings of ISCC 2000*, pp. 214–219, July 2000.

[15] Guerin, R., Ahmadi, H., and Naghshineh, M., "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 968–981, Sept. 1991.

[16] Guerin, R., Orda, A., and Williams, D., "QoS routing mechanisms and OSPF extensions," in *Proceedings of GLOBECOM 1997*, pp. 1903–1908, Nov. 1997.

[17] Heinanen, J., Baker, F., Weiss, W., and Wroclawski, J., "Assured forwarding PHB group." IETF RFC 2597, June 1999.

[18] Jacobson, V., "Congestion avoidance and control," in *Proceedings of ACM SIGCOMM 1988*, pp. 314 – 329, Aug. 1988.

[19] Jamoussi, B., Andersson, L., Callon, R., Dantu, R., Wu, L., Doolan, P., Worster, T., Feldman, N., Fredette, A., Girish, M., Gray, E., Heinanen, J., Kilty, T., and Malis, A., "Constraint-based LSP setup using LDP." IETF RFC 3212, Jan. 2002.

[20] Katsube, Y., Nagami, K., and Esaki, H., "Toshiba's router architecture extensions for ATM : Overview." IETF RFC 2098, Feb. 1997.

[21] Kelly, F., Maulloo, A., and Tan, D., "Rate control in communication networks: shadow prices, proportional fairness and stability," in *Journal of the Operational Research Society*, vol. 49, 1998.

[22] Kleinberg, J. M., Rabani, Y., and Tardos, E., "Fairness in routing and load balancing," in *IEEE Symposium on Foundations of Computer Science*, pp. 568–578, 1999.

[23] Kodialam, M. and Lakshman, T., "Minimum interference routing with applications to MPLS traffic engineering," in *Proceedings of INFOCOM 2000*, pp. 884–893, Mar. 2000.

[24] Koehler, B. G., *Best-effort traffic engineering in multiprotocol label switched networks*. PhD dissertation, Georgia Institute of Technology, 2002.

[25] Laubach, M. and Halpern, J., "Classical IP and ARP over ATM." IETF RFC 2225, Apr. 1998.

[26] Lee, W. C., Hluchyi, M. G., and Humblet, P. A., "Routing subject to quality of service constraints in integrated communication networks," *IEEE Network*, pp. 46–55, July 1995.

[27] Lin, Y.-D. J., Hsu, N.-B., and Hwang, R.-H., "QoS routing granularity in MPLS networks," *IEEE Communications Magazine*, pp. 58–65, June 2002.

[28] Luciani, J., Katz, D., Piscitello, D., Cole, B., and Doraswamy, N., "NBMA next hop resolution protocol (NHRP)." IETF RFC 2332, Apr. 1998.

[29] Ma, Q. and Steenkiste, P., "On path selection for traffic with bandwidth guarantees," in *Proceedings of ICNP 1997*, pp. 191–202, Oct. 1997.

[30] Malkin, G., "RIP version 2." IETF RFC 2453, Nov. 1998.

[31] Mo, J. and Walrand, J., "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556–567, 2000.

[32] Moy, J., "OSPF version 2." IETF RFC 2328, Apr. 1998.

[33] Newman, P., Minshall, G., and Lyon, T. L., "IP switching - ATM under IP," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 117 – 129, Apr. 1998.

[34] Postel, J., "User datagram protocol." IETF RFC 768, Aug. 1980.

[35] Postel, J., "Transmission control protocol." IETF RFC 793, Sept. 1981.

[36] Rekhter, Y., Davie, B., Katz, D., Rosen, E., and Swallow, G., "Cisco systems' tag switching architecture overview." IETF RFC 2105, Feb. 1997.

[37] Roberts, J., "Flow aware networking for effective quality of service control," in *IMA Workshop on Scaling*, Oct. 1999.

[38] Rojanarowan, J., Koehler, B., and Owen, H., "MPLS based best effort traffic engineering," in *Proceedings of SECON 2004*, pp. 239–245, Mar. 2004.

[39] Rojanarowan, J., Koehler, B., and Owen, H., "Traffic engineering using MPLS for best effort traffic," in *Proceedings of ICN 2004*, pp. 84–89, Mar. 2004.

[40] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and Conta, A., "MPLS label stack encoding." IETF RFC 3032, Jan. 2001.

[41] Rosen, E., Viswanathan, A., and Callon, R., "Multiprotocol label switching architecture." IETF RFC 3031, Jan. 2001.

[42] Shenker, S., Partridge, C., and Guerin, R., "Specification of guaranteed quality of service." IETF RFC 2212, Sept. 1997.

[43] Shreedhar, M. and Varghese, G., "Efficient fair queueing using deficit round robin," in *Proceedings of ACM SIGCOMM 1995*, pp. 231–242, 1995.

[44] Subramanian, S. and Muthukumar, V., "Alternate path routing algorithm for traffic engineering in the internet," in *Proceedings of ITCC 2003*, pp. 367–372, Apr. 2003.

[45] Suri, S., Waldvogel, M., , and Warkhede, P., "Profile-based routing: A new framework for MPLS traffic engineering," in *Proceedings of QofIS 2001*, pp. 138–157, Sept. 2001.

[46] Vutukury, S. and Garcia-Luna-Aceves, J. J., "A traffic engineering approach based on minimum-delay routing," in *Proceedings of ICCCN 2000*, pp. 16–19, Oct. 2000.

[47] Wang, Y. and Wang, Z., "Explicit routing algorithms for internet traffic engineering," in *Proceedings of ICCCN 1999*, pp. 582–588, Oct. 1999.

[48] Wang, Z. and Crowcroft, J., "Quality-of-service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1228–1234, Sept. 1996.

[49] Woundy, R., Viswanathan, A., Feldman, N., , and Boivie, R., "ARIS: Aggregate route-based IP switching." IETF Internet Draft, Nov. 1996.

[50] Wroclawski, J., "Specification of the controlled-load network element service." IETF RFC 2211, Sept. 1997.

[51] Wydrowski, B. and Zukerman, M., "GREEN: An active queue management algorithm for a self managed internet," in *Proceedings of ICC 2002*, pp. 2368–2372, Apr. 2002.

[52] Wydrowski, B. and Zukerman, M., "QoS in best-effort networks," *IEEE Communications Magazine*, pp. 44–49, Dec. 2002.

[53] Zegura, E. W., Calvert, K. L., and Bhattacharjee, S., "How to model an internetwork," in *Proceedings of INFOCOM 1996*, pp. 594–602, Mar. 1996.

VITA

Jerapong Rojanarowan was born in Petchaboon, Thailand in 1972. He received his B.Eng. and M.Eng. degrees, both in Electrical Engineering, from Chulalongkorn University, Bangkok, Thailand. He was awarded a scholarship from Assumption University, Bangkok, Thailand to pursue a Ph.D. degree at the Department of Electrical and Computer Engineering at Georgia Institute of Technology in January, 2002. In December, 2005, he obtained his Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology. His research interests include traffic engineering, bandwidth allocation, routing, and optimization.